

26845/14106



**PENERAPAN ALGORITME SEGMENTASI MEAN SHIFT  
DAN PEMILIHAN FITUR DASAR WARNA  
PADA SISTEM TEMU KEMBALI CITRA BERBASIS ISI**

**TUGAS AKHIR**

PSI f  
005.1  
Hid  
D-1  
2006



PERPUSTAKAAN ITS	
Tgl. Terima	15-8-06
Terima Dari	H
No. Agenda Prp.	27.6421

Disusun Oleh :

**NIZAR IHROMI HIDAYAT**  
5102 100 032

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2006**

**PENERAPAN ALGORITME SEGMENTASI MEAN SHIFT  
DAN PEMILIHAN FITUR DASAR WARNA  
PADA SISTEM TEMU KEMBALI BERBASIS ISI**

**TUGAS AKHIR**

**Diajukan untuk Memenuhi Sebagian Persyaratan  
Memperoleh Gelar Sarjana Komputer  
Pada  
Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya**

**Mengetahui / Menyetujui**

**Dosen Pembimbing I**



**Rully Soelaiman, S.Kom, M.Kom**  
**NIP. 132085802**

**Dosen Pembimbing II**



**Nanik Suciati, S.Kom, M.Kom**  
**NIP. 132125674**

**SURABAYA  
AGUSTUS, 2006**





Bismillah...

ABSTRAK



## ABSTRAK

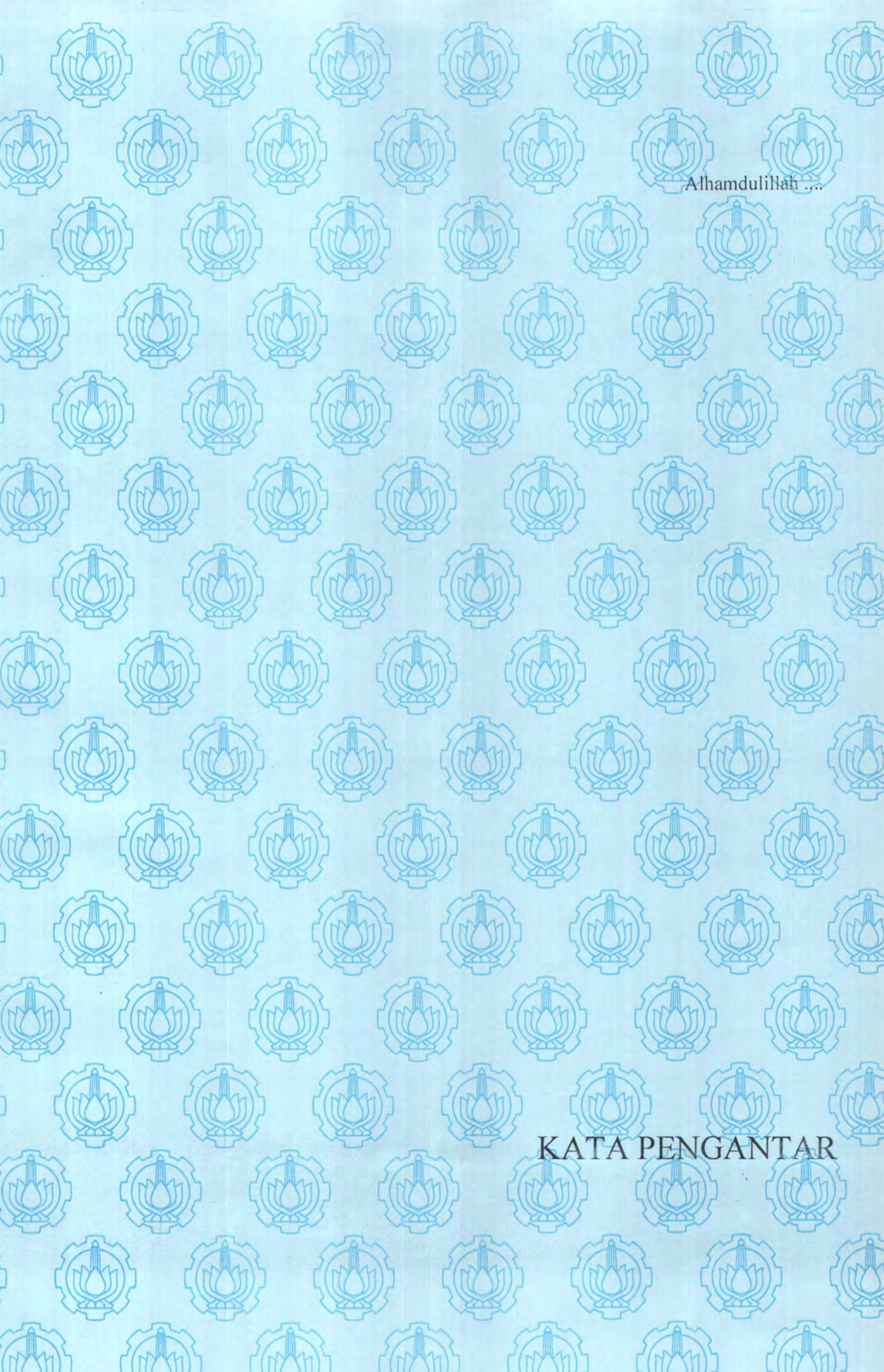
*Sistem Temu Kembali Citra Berbasis Isi adalah sistem yang mengorganisir data citra hasil pengenalan objek, mengenali data citra baru dan mampu melakukan pencarian data citra berdasarkan objek dari citra dalam jumlah citra besar dan variasi gambar yang beragam. Salah satu implementasi penggunaan Sistem Temu Kembali Citra Berbasis Isi adalah pencarian koleksi barang di museum. User dapat memperoleh informasi gambar tentang barang-barang koleksi museum hanya dengan memasukkan nama objek dari barang koleksi.*

*Pada Tugas Akhir ini akan difokuskan pertama pada pembuatan Sistem Temu Kembali Citra Berbasis Isi dengan tiga tahapan. Pada tahapan pertama pelatihan data, data citra yang telah diberi label/keyword berisikan nama-nama objek pada citra disegmentasi menggunakan algoritme Mean Shift untuk menghasilkan region-region. Untuk menghasilkan vektor fitur citra maka dilakukan ekstraksi fitur dasar warna pada hasil segmentasi, mengelompokkan/clustering region ke dalam kelas-kelas/blob-blob dengan menggunakan algoritme K-Means berdasarkan vektor fitur, memprediksi keterhubungan antara blob dengan word dengan menggunakan algoritme Expectation Maximization. Tahap kedua, Pengenalan Data Citra baru dilakukan dengan menggunakan algoritme Nearest-Neighbour berdasarkan kedekatan vektor fitur data citra baru dengan data citra hasil pelatihan. Pada tahap ketiga dilakukan Pencarian Data Citra berdasarkan data input user berupa teks yang mewakili nama objek yang dicari dari data citra yang telah dihasilkan pada tahapan pelatihan data. Fokus kedua yaitu analisis kinerja Sistem Temu Kembali Citra Berbasis Isi hasil dari pemilihan fitur dasar warna dalam domain ( $L^*u^*v$ , RGB,  $L^*a^*b$ ,  $L^*a^*b$  dan CrCb), penggunaan algoritme mean shift pada tahapan awal/preprocessing.*

*Berdasarkan hasil uji coba, penentuan jumlah klaster yang lebih banyak terbukti meningkatkan kinerja Sistem Temu Kembali Citra Berbasis Isi serta pemilihan fitur dasar warna domain  $L^*u^*v$  lebih memiliki nilai kinerja yang lebih tinggi daripada domain warna yang lain ( $L^*u^*v$ , RGB,  $L^*a^*b$ ,  $L^*a^*b$ , CrCb, dan kombinasinya)*

**Kata Kunci:** *Clustering, Expectation Maximization, K-Means, Mean Shift Nearest-Neighbour*





Alhamdulillah ....

## KATA PENGANTAR



## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadapan Allah Subhanahu wata'ala. Karena limpahan rahmat dan karunia-Nyalah penulis dapat menyelesaikan Tugas Akhir yang berjudul **“Penerapan Algoritme Segmentasi *Mean Shift* dan Pemilihan Fitur Dasar Warna pada Sistem Temu Kembali Citra Berbasis Isi”** . Semoga sholawat dan salam tetap tercurahkan kepada junjungan Nabi Muhammad SAW.

Tugas Akhir ini dibuat guna memenuhi persyaratan akademik dalam rangka ujian akhir bagi mahasiswa Strata 1 (S1) Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember, Surabaya.

Dalam Tugas Akhir ini penulis mengetengahkan pengembangan sistem temu kembali citra berbasis isi.

Dalam penyusunan Tugas Akhir ini, Penulis berusaha untuk menerapkan ilmu yang telah didapat selama menjalani perkuliahan dengan tidak terlepas dari petunjuk, bimbingan, bantuan, dan dukungan berbagai pihak. Pada kesempatan ini penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

- Bapak dan ibu di rumah, kakak atas didikan selama ini.
- Keluarga di Cianjur, Surabaya, Bondowoso, Depok, Kediri, dan Jember.
- Bapak Rully Soelaiman, S.Kom, M.Kom selaku dosen pembimbing utama.



- Ibu Nanik Suciati, S.Kom., M.Kom selaku dosen pembimbing kedua dan dosen wali.
- Bapak M Nuh DEA, Bapak Ahmad Jazidie selaku pimpinan ITS.
- Bapak Arif Djunaidy, Bapak Yudhi Purwananto selaku pimpinan FTIF.
- Bung Karno, Bung Hatta, Pak Mahmud Ahmadinejad, pak SBY, pak Hidayat Nurwahid, Ustad Musholli, pak Mutashor, pak Fatkur, pak Munir, Kamal 00, Nugroho 98, Pak Guntar 97 sebagai guru spiritual.
- Bu Esther Hanaya, bu Bilqis, bu Mediana selaku dosen penguji.
- Pak Ary, Pak Darlis, bu Chastine, bu Umy dan seluruh Dosen Teknik Informatika ITS atas kesediaan membagi ilmunya kepada penulis.
- Seluruh Karyawan Teknik Informatika ITS : mas Yudhi, mbak Fatin, mbak Eva, Pak Kar atas kerjasamanya.
- Saudara-saudara seperjuangan Toni, Haris, Saud, Andri, Ihwana, Feby.
- Hain, Knip, Daniel, Huda, Henning, Hendra, Issa, Andy S, Aini, Tedy, Purna, Eko, Mbah Rendi, Dimas, Amiril, Dedy, Anti, Frans, Roy, Heri, dan saudara-saudara seangkatan 2002 yang lain.
- Ziyad, Chasby, Farid, Heru, Arya, Lukman, Wid, Wahyu ,Andi ,Cesar, Ustad, Moko, Gie, Ali A, Wafa, serta teman-teman pejuang PPSDMS.
- Pak Rendra, mbak Renni, pak Gatot, Pak Martono, Pak Detak, Agis F, Mas Salim, Mbak Akhada, Rio, Tatit, Indra, Heni, Ita, Adit, Hikmah, Fauzan, Siti, Muflihah, Yudha, Fajar, Hendri serta teman-teman BEM ITS.
- Ninik, Erik, Aji, Inge', Adi BK 'Kantin', Mas Indro, Mas Yeka, Lini serta teman-teman pejuang ITS.




- Adik Rile, Frandy, Ucup, Yasrif, Evi, Ratih, Karlina, Belina, Dwi, Imam, Firman serta adik-adik 2003 yang lain.
- Adik Aris, Tanto, Pinky, Fatih, Novia, Novita, Licha serta adik-adik 2004 yang lain.
- Mas Mardi01, mas Udin00, mas Lintang00, mas Cahyo 97, mbak Putri00, mbak Ningrum99, mas Adhe99, mas Kurniawan00 serta kakak-kakak yang lain.
- Kang Fauzan00, Kang Mahmud00, Mas Gayuh, Fahru05, serta seluruh admin dan penghuni Laboratorium Komputing, Laboratorium Pemrograman, Laboratorium AJK, Laboratorium RPL, Laboratorium IBS.
- Mahe, Agung, Wignyot serta teman teman sekos-an GK 40.
- Mak Ning, Mak Di serta pemilik-pemilik warung di sekitar ITS.

Bagaimanapun juga Penulis telah berusaha sebaik-baiknya dalam menyusun Tugas Akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Penulis mengharapkan adanya kritik dan saran yang membangun guna menambah manfaat serta mengurangi kesalahan dan kekurangan yang ada. Pada akhirnya penulis berharap semoga laporan ini dapat memberikan manfaat bagi kita semua.

Surabaya, 24 Juli 2006

Nizar Ihromi Hidayat





Maka apabila langit telah terbelah dan  
Menjadi **MERAH MAWAR**  
seperi kilapan minyak  
Maka nikmat Tuhan kamu  
yang manakah yang kamu dustakan

**QS. Ar Rahman : 37-38**

**DAFTAR ISI**



# DAFTAR ISI

<b>ABSTRAK.....</b>	<b>IV</b>
<b>KATA PENGANTAR .....</b>	<b>V</b>
<b>DAFTAR ISI.....</b>	<b>VIII</b>
<b>DAFTAR GAMBAR .....</b>	<b>XI</b>
<b>DAFTAR TABEL .....</b>	<b>XIII</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>1</b>
1.1 LATAR BELAKANG.....	1
1.2 PERMASALAHAN .....	5
1.3 TUJUAN PEMBUATAN TUGAS AKHIR .....	6
1.4 BATASAN MASALAH.....	8
1.5 METODOLOGI PEMBUATAN.....	8
1.6 SISTEMATIKA PENULISAN .....	9
<b>BAB 2 SEGMENTASI MEAN SHIFT .....</b>	<b>11</b>
2.1 DASAR TEORI MEAN SHIFT .....	11
2.1.1 Multivariate kernel density.....	12
2.1.3 Diferensiasi Kernel .....	12
2.1.4 Diferensiasi Kernel Epanichnikov .....	13
2.1.5 Konvergensi Mean Shift .....	15
2.2 PENERAPAN ALGORITMA MEAN SHIFT PADA SEGMENTASI CITRA.....	20
2.2.1 Mean Shift Filter .....	21
2.2.3 Segmentasi Mean Shift .....	23
<b>BAB 3 SISTEM TEMU KEMBALI CITRA BERBASIS ISI .....</b>	<b>25</b>
3.1 PELATIHAN DATA CITRA.....	25
3.1.1 Segmentasi Citra.....	26
3.1.2 Ekstraksi Fitur Dasar Warna .....	26
3.1.2.1 Fitur Dasar Centroid/pusat massa, Area-percontour .....	26
3.1.2.2 Fitur Mean dan Varian .....	27
3.1.3 Pengelompokan vektor fitur dari setiap region dengan Algoritme K- Means.....	31
3.1.3.1 Dasar Teori Algoritme K-Means.....	31
3.1.3.2 Pengelompokan data Region dengan algoritme K Means .....	32
3.1.4 Menghubungkan Blob dengan Word dengan Algoritme Expectation Maximization.....	34
3.2 PENGENALAN DATA CITRA BARU .....	54
3.3 PENCARIAN DATA CITRA .....	55
3.3.1 Organisasi Data.....	55
3.3.2 Strategi Pencarian .....	56
<b>BAB 4 PERANCANGAN SISTEM .....</b>	<b>58</b>
4.1 PEMODELAN DESAIN DATA.....	58
4.1.1 Desain Data CDM.....	58
4.1.2 Desain Data PDM.....	60
4.2 PEMODELAN PROSES.....	62
4.2.1 DFD Level 0 Proses Utama dari CBIR.....	62
4.2.2 DFD Level 1.....	63



4.2.2 DFD Level 2.....	64
4.2.2.1 DFD Level 2 Pelatihan Data Citra.....	64
4.2.2.2 DFD Level 2 Pengenalan Objek Data Citra Baru.....	66
4.3 PEMODELAN DIAGRAM ALUR ALGORITME.....	67
4.3.1 Algoritme Ekstraksi Fitur Dasar dan Warna.....	68
4.3.2 Algoritme Mean Shift.....	69
4.3.3 Algoritme K-Means.....	69
4.3.4 Algoritme Nearest-Neighbor.....	71
4.3.5 Algoritme Expectation Maximization.....	71
4.4 PEMODELAN ANTAR MUKA.....	73
<b>BAB 5 IMPLEMENTASI SISTEM .....</b>	<b>74</b>
5.1 IMPLEMENTASI MEAN SHIFT.....	74
5.2 IMPLEMENTASI EKSTRAKSI FITUR.....	76
5.2.1 Fitur Dasar.....	76
5.2.1 Mean dan Varian.....	76
5.2.1.1 nilai RGB.....	76
5.2.1.2 Domain CIE XYZ.....	76
5.2.1.3 Domain CrCb.....	76
5.2.1.4 Domain $L^*u^*v$ .....	77
5.2.1.5 Domain CIE Lab.....	77
5.3 IMPLEMENTASI KLASTERISASI K MEANS.....	78
5.4 IMPLEMENTASI EXPECTATION MAXIMIZATION.....	78
5.4.1 Inisialisasi $t$ dan $p(a_{nj}=i)$ .....	78
5.4.2 Normalisasi $t$ .....	79
5.4.3 Tahap E.....	79
5.4.4 Tahap M.....	80
5.4.5 Konvergen.....	81
5.4.6 Penerapan EM.....	82
5.4.6.1 Inisialisasi $p(a_{nj}=i)$ .....	83
5.4.6.2 Inisialisasi $t$ .....	85
5.4.6.3 Tahap E.....	86
5.4.6.4 Tahap M.....	89
5.5 IMPLEMENTASI NEAREST-NEIGHBOR.....	92
<b>BAB 6 UJI COBA DAN EVALUASI .....</b>	<b>93</b>
6.1 DATA INPUT.....	93
6.1.1 Penggunaan Corel Data Set sebagai data input.....	93
6.1.2 Informasi Word pada data input.....	94
6.2.3 Evaluasi Segmentasi.....	96
6.2.4 Ekstraksi Fitur Dasar Warna.....	97
6.2.5 Klasterisasi Region.....	97
6.2 METODE EVALUASI.....	100
6.2.1 Evaluasi Strategi Region Naming.....	100
6.2.2 Evaluasi Strategi Auto-Annotation.....	101
6.2.2.1 Skor Normalized Clasification.....	101
6.2.2.2 Recall dan precision.....	102
6.2.2.3 True Positive, False Positive, False Negative.....	102
6.3 SKENARIO UJI COBA PARAMETER JUMLAH KELAS/BLOB.....	103
6.4 SKENARIO UJI COBA PARAMETER FITUR DASAR WARNA.....	103
6.4.1 Uji coba 1.....	104
6.4.1.1 Pengukuran Skor normalized clasification.....	104
6.4.1.2 Pengukuran Recall/precssion.....	105
6.4.1.3 Pengukuran true postife, word predict, false positif dan false negative.....	106
6.4.2 Uji coba 2.....	108
6.4.2.1 Pengukuran Skor normalized clasification.....	108
6.4.2.2 Pengukuran Recall/precssion.....	109

6.4.2.3 Pengukuran true postive, word predict , false positive dan false negative.....	110
<b>BAB 7 KESIMPULAN DAN SARAN.....</b>	<b>114</b>
7.1 KESIMPULAN .....	114
7.2 SARAN.....	114
<b>DAFTAR PUSTAKA .....</b>	<b>116</b>



## DAFTAR GAMBAR

Gambar 1.1 Blob World.....	3
Gambar 1.2 Tahapan CBIR 1 .....	7
Gambar 2.3 Density Estimation.....	12
Gambar 2.4 Perpindahan vektor Mean Shift ke arah kerapatan maksimal lokal dari perhitungan mean shift .....	15
Gambar 2.5 Jendela <i>Mean Shift</i> .....	17
Gambar 2.6 Ilustrasi Iterasi <i>Mean Shift</i> Iterasi 1-6 .....	22
Gambar 2.7 Ilustrasi Iterasi <i>Mean Shift</i> Iterasi 7-9 .....	23
Gambar 3.8 Kesamaan Citra dari Beberapa Citra.....	33
Gambar 3.9 <i>clustering region</i> .....	33
Gambar 3.10 keterhubungan <i>blob</i> dengan <i>word</i> .....	34
Gambar 3.11 Contoh Distribusi <i>Mixture Model</i> .....	36
Gambar 3.12 Peluang Keterhubungan <i>Word</i> dan Blob.....	50
Gambar 3.13 <i>Region Naming</i> .....	53
Gambar 3.14 <i>Auto-Annotation</i> .....	54
Gambar 3.15 Kelas Objek Gedung , Langit, Burung pada Citra .....	56
Gambar 3.16 Pencarian pada Kelas Objek Gedung dan Burung .....	57
Gambar 4.17 Conceptual Data Model .....	59
Gambar 4.18 Physical Data Model.....	60
Gambar 4.19 DFD Level 0 .....	62
Gambar 4.20 DFD Level 1 .....	63
Gambar 4.21 DFD Level 2 .....	64
Gambar 4.22 DFD Level 2 .....	66
Gambar 4.23 Algoritme Ekstraksi Fitur .....	68
Gambar 4.24 Algoritme Segmentasi <i>Mean Shift</i> .....	69
Gambar 4.25 Algoritme K-Means .....	70
Gambar 4.26 algoritme Nearest Neighbor .....	71
Gambar 4.27 Algoritma EM .....	72

Gambar 4.28 Antarmuka perangkat lunak .....	73
Gambar 5.29 Citra Penerapan EM.....	82
Gambar 5.30 Keterhubungan <i>Blob</i> dan <i>Word</i> dalam Suatu Citra.....	83
Gambar 6.31 Contoh Data <i>Input Corel Data Set</i> .....	94
Gambar 6.32 Frekuensi <i>Word</i> dalam Setiap <i>Blob</i> .....	95
Gambar 6.33 Frekuensi <i>Word</i> dalam Keseluruhan Citra.....	95
Gambar 6.34 Perbandingan Hasil Segmentasi <i>Mean Shift</i> .....	96
Gambar 6.35 Hasil Segmentasi <i>Mean Shift</i> .....	96
Gambar 6.36 perbandingan klaster antara 5 dataset .....	98
Gambar 6.37 <i>Blob</i> Segmentasi.....	99
Gambar 6.38 Hasil strategi <i>region naming</i> .....	100
Gambar 6.40 <i>Recall</i> dan <i>Precision</i> dari dataset 1-6 .....	105
Gambar 6.41 <i>Recall</i> dan <i>Precision</i> dari dataset 7 .....	106
Gambar 6.42 <i>Recall</i> dan <i>Precision</i> dari dataset 1-6.....	109
Gambar 6.43 <i>Recall</i> dan <i>Precision</i> dari dataset 7 .....	110





DAFTAR TABEL



## DAFTAR TABEL

Tabel 4.1 Keterangan CDM.....	59
Tabel 4.2 Keterangan Phisycal Data Model .....	61
Tabel 5.3 peluang korespondensi blob pada citra 1 sampai 6 .....	84
Tabel 5.4 peluang korespondensi <i>blob</i> pada citra 1 - 6 dengan <i>word</i> ke-2.....	84
Tabel 5.5 peluang korespondensi <i>blob</i> pada citra 1 sampai 6 dengan <i>word</i> ke- 3.....	85
Tabel 5.6 intensitas kehadiran <i>blob</i> -i dengan <i>word</i> -j.....	85
Tabel 5.7 Normalisasi t.....	86
Tabel 5.8 p_theta <i>blob</i> -i dengan <i>word</i> -1 pada citra 1 sampai 8.....	86
Tabel 5.9 p_theta <i>blob</i> -i dengan <i>word</i> -2 pada citra 1 sampai 8.....	87
Tabel 5.10 p_theta <i>blob</i> -i dengan <i>word</i> -2 pada citra 1 sampai 8.....	87
Tabel 5.11 normalisasi p_theta <i>blob</i> -i dengan <i>word</i> -1 pada citra 1 -8 .....	88
Tabel 5.12 p_theta blob -1 dengan <i>word</i> -3 pada citra 1 – 8 .....	88
Tabel 5.13 normalisasi p_theta <i>blob</i> -i dengan <i>word</i> -1 pada citra 1 – 8.....	89
Tabel 5.14 keterhubungan <i>blob</i> dan <i>word</i> pada citra 1 sampai 8 .....	89
Tabel 5.15 normalisasi <i>blob</i> -i dengan <i>word</i> -1 .....	90
Tabel 5.16 nilai t pada citra 1 sampai 8 .....	90
Tabel 5.17 tabel t .....	91
Tabel 5.18 normalisasi t blob-i dengan <i>word</i> -j pada citra 1 – 8 .....	91
Tabel 6.19 Data set Input.....	97
Tabel 6.20 Perbandingan iterasi dengan nilai <i>maximum likelihood</i> .....	103
Tabel 6.21 Kamus kata dengan 14 keyword.....	104
Tabel 6.22 perbandingan <i>normalized clasifcation</i> skor .....	104
Tabel 6.23 <i>true positive, word predict , false positive dan false negative data set</i> 1-4 .....	106
Tabel 6.24 <i>true positive, word predict , false positive dan false negative data set</i> 5-7 .....	107
Tabel 6.25 Kamus kata dengan 18 keyword .....	108
Tabel 6.26 perbandingan <i>normalized clasifcation</i> skor .....	108
Tabel 6.27 true postive, word predict , false positive dan false negative .....	111



Tabel 6.28 true postive, word predict , false positive dan false negative .....112

betapa inginnya agar kami bangsa ini mengetahui bahwa  
mereka lebih kami cintai daripada diri kami sendiri  
kami berbangga ketika jiwa jiwa kami gugur  
sebagai penebus bagi kehormatan mereka  
jika memang tebusan itu yang diperlukan  
atau menjadi harga bagi tegaknya bagi kejayaan,  
kemuliaan dan terwujudnya cita cita mereka  
jika memang itu harga yang harus dibayar  
tiada sesuatu yang membuat kami bersikap seperti ini  
selain rasa cinta yang telah mengharu biru  
menguasai perasaan kami  
memeras habis air mata kami  
dan mencabut rasa ingin tidur dari pelupuk mata kami

**IDEALISME KAMI-I PPSDMS**

## **BAB I PENDAHULUAN**



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

*Computer vision* [1] adalah ilmu dan metode aplikasi dalam menggunakan komputer untuk memahami isi citra (*image content*). Area permasalahan dalam *computer vision* adalah pengukuran dan pemrosesan, yang dapat dilakukan dengan berbagai metode. Beberapa area permasalahan *computer vision* adalah sebagai berikut:

1. *Recognition*/Pengenalalan bertujuan mengenali objek data citra, aplikasinya seperti *Content Based Image Retrieval (CBIR)*, *Optical Character Recognition (OCR)*.
2. *Motion*/gerakan bertujuan mengenali data citra bergerak. Aplikasinya seperti *Egomotion* yang membagi gerakan 3D dari kamera, *Tracking* yang memperkirakan satu atau beberapa objek dalam citra bergerak
3. Restorasi citra, bertujuan untuk mendapatkan data citra, citra bergerak atau objek 3D tanpa *noise*.

*Content Based Image Retrieval* (Temu Kembali Citra Berbasis Isi) adalah salah satu aplikasi dari *computer vision* dalam permasalahan temu kembali citra dalam jumlah data citra yang besar. Arti "Berbasis Isi" adalah pencarian citra berdasarkan isi citra. Terminologi CBIR dikenalkan pada tahun 1992 oleh T.Kato, dalam ujicobanya dijelaskan sistem temu kembali citra dari *database* citra



berdasarkan warna dan bentuk. Sejak itu, terminologi CBIR digunakan dalam menjelaskan proses temu kembali citra yang diinginkan dari koleksi citra yang besar. Sistem CBIR yang ideal dalam prespektif user adalah pencarian citra berdasarkan teks.

Hingga saat ini telah dikembangkan beberapa sistem temu kembali citra berbasis isi dalam berbagai aplikasi sebagai berikut :

- *Query By Image Content* (QBIC) dikembangkan oleh IBM. Dalam QBIC, *query*/permintaan user bisa dilakukan dengan menggunakan citra contoh, sket yang digambar user, atau tekstur dan warna yang dipilih *user*. Demo *on-line* bisa dilihat di <http://www.qbic.ibm.com>.
- *Netra* dikembangkan di UC Santa Barbara. Informasi mengenai warna, tekstur, bentuk dan lokasi dari *region* (bagian-bagian citra yang memiliki ciri-ciri yang unik) citra digunakan untuk menemukan kembali *region-region* yang mirip dari citra-citra yang tersimpan dalam *database*(sistem penyimpanan dan pengelolaan data). Demo *on-line* bisa dilihat di <http://maya.ece.ucsb.edu/Netra> .
- *Blobworld* dikembangkan di UC Berkeley. Tiap citra disegmentasi secara otomatis ke dalam *region* berdasarkan fitur warna dan tekstur. *Query* dilakukan dengan menggunakan atribut-atribut dari satu atau dua *region* yang diinginkan (bersifat lokal), tidak menggunakan atribut-atribut global yang ada dalam satu citra. Demo *on-line* bisa dilihat di <http://elib.cs.berkeley.edu/vision.html> .





**Gambar 1.1 Blob World**

- *Retrieval Ware* dikembangkan oleh Excalibur Technologies Corp. Pencarian pada *Retrieval Ware* dilakukan berdasarkan fitur warna , bentuk, tekstur isi, kecerahan, struktur warna, dan rasio. *Retrieval Ware* mendukung kombinasi dari fitur-fitur tersebut dengan nilai fitur yang dapat dikontrol oleh user. Demo *on-line* bisa dilihat di <http://vrw.excalib.com>.
- *Image Rover* dikembangkan oleh Boston University, mengkombinasikan fitur tekstual dan visual untuk pencarian citra di web. User mengawali pencarian dengan memberikan teks yang spesifik dari citra yang diharapkan. Demo *on-line* bisa dilihat di <http://cs.-www.bu.edu/groups/ivc/ImageRover> .
- MARS dikembangkan di University of Illinois at Urbana-Champaign. MARS mengolah berbagai fitur visual ke dalam arsitektur *retrieval* yang

berguna dan dapat digunakan pada aplikasi dan user yang berbeda. Demo *on-line* bisa dilihat di <http://www-db.ics.uci.edu/pages/demos> .

- PicToSeek adalah *image retrieval* untuk web, dikembangkan oleh ISIS Research Group di Univeristy of Amsterdam. Ide utamanya adalah mengekstrak *invariant fitur* (fitur yang berlainan) dari setiap citra pada *database*, kemudian hasil ekstraksi disesuaikan dengan kumpulan fitur *invariant* dari citra *query*. Demo *on-line* bisa dilihat di <http://www.science.uva.nl/research/izis/zomax> .

Pada sistem-sistem yang sudah dikembangkan sebelumnya, kebanyakan *query* dilakukan dengan menggunakan sket, citra contoh, atau region contoh. Penggunaan *query* ini kurang memudahkan user. Seringkali user ingin melakukan *query* dengan meng-*input*-kan deskripsi teks mengenai “objek-objek” yang ada dalam citra. User menginputkan teks ”orang”, ”perahu”, ”laut”, jika user ingin mencari bermacam-macam citra yang bertema laut dan di dalamnya juga memuat orang dan perahu.

Untuk memenuhi kebutuhan tersebut, dapat dilakukan pengindeksan secara manual, artinya sebelum citra disimpan ke dalam database, objek-objek yang terdapat dalam citra tersebut dikenali secara manual terlebih dahulu oleh seorang operator, setelah itu baru disimpan ke dalam database berdasarkan objek-objek yang sudah dikenali. Jika ukuran citra sangat besar, pengindeksan manual akan sulit dilakukan, karena:

- Membutuhkan waktu yang sangat lama



- Bersifat subyektif, sebuah gambar bisa dimaknai berbeda oleh dua orang operator

Oleh karena itu diperlukan suatu sistem pengenalan objek otomatis untuk keperluan pengorganisasian data citra berukuran besar. Penggunaan CBIR Tiga tahapan Sistem Temu Kembali Citra berbasis isi yaitu pelatihan data, pengenalan objek, dan pencarian data. Skema dari sistem yang dibangun terlihat pada Gambar 1.2.

## 1.2 Permasalahan

Permasalahan yang dihadapi dalam Tugas Akhir ini adalah :

- Bagaimana mengimplementasikan metode segmentasi citra dengan algoritme *Mean Shift* [2].
- Bagaimana mengekstrak fitur-fitur dasar dari *region-region* hasil segmentasi. Fitur yang dimaksud adalah sebagai berikut : ukuran *region*; lokasi *region*; luas *region* dibagi akar dari panjang *contour region*; *mean* dan *varian dalam berbagai domain warna*.
- Bagaimana melakukan proses *clustering region* dari fitur-fitur hasil ekstraksi dengan menggunakan algoritme *K-Means*. Hasil dari proses ini adalah *region-region* dari keseluruhan citra terkelompok dalam dalam kelas-kelas tertentu. Kelas tersebut dinamakan *blob*.
- Bagaimana mengimplementasikan algoritme *Expectation Maximization* (EM) [5] dalam pembuatan tabel peluang untuk menentukan hubungan

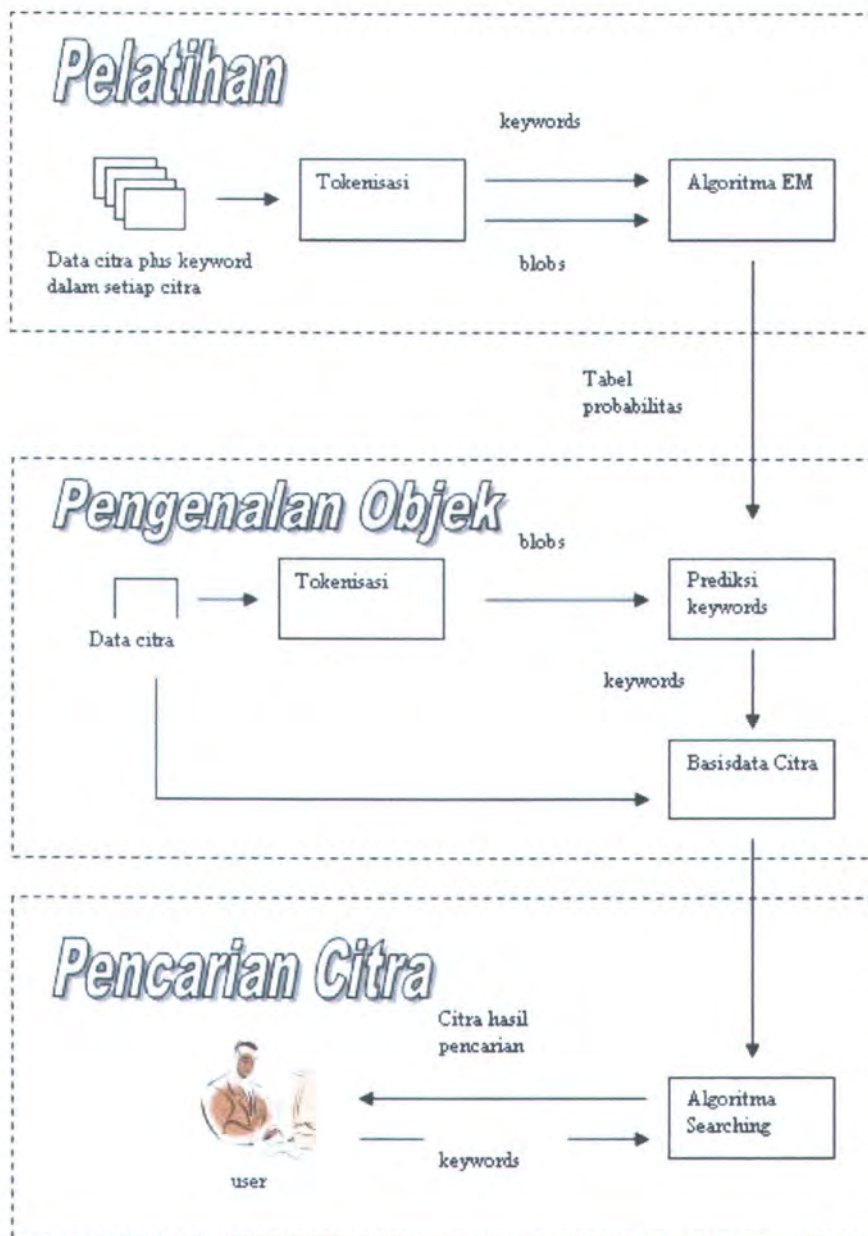
antara *blob* dan *word*. *Word* adalah kata yang menyimpan nama-nama objek dari keseluruhan citra, *word* tersimpan dalam kamus kata (*lexicon*).

- Bagaimana membuat desain alur sistem, struktur data, algoritme fungsi serta antarmuka perangkat lunak yang akan diterapkan dalam tahap implementasi sistem temu kembali citra berbasis isi.
- Bagaimana melakukan pengorganisasian dan penyimpanan data citra yang akan digunakan dalam sistem temu kembali citra berbasis isi.
- Bagaimana melakukan analisis hasil sistem temu kembali citra untuk melihat kinerja dari penggunaan metode yang dipilih dalam pembuatan sistem temu kembali citra berbasis isi.

### 1.3 Tujuan Pembuatan Tugas Akhir

Tujuan pembuatan Tugas Akhir ini adalah membuat suatu perangkat lunak yang mampu mengorganisir data citra dalam beragam citra dan jumlah yang besar dari hasil pengenalan objek pada tahap pelatihan data, melakukan pengenalan objek dari data citra baru, dan melakukan proses temu kembali citra berbasis isi dengan data *input user* berupa teks yang mewakili nama objek yang diinginkan. Tujuan kedua yaitu melakukan analisis pada kinerja hasil sistem temu kembali citra terhadap penerapan algoritme mean shift pada tahap *pre-processing* dan pemilihan fitur dasar warna dalam domain warna yang berbeda





Gambar 1.2 Tahapan CBIR 1

#### 1.4 Batasan Masalah

Sejumlah permasalahan yang dibahas dalam Tugas Akhir ini akan dibatasi ruang lingkup pembahasannya, antara lain :

- Data input dalam pencarian citra berupa teks. Teks adalah nama objek yang diinginkan user dari *database* citra.
- Setiap citra tersimpan 2-3 *keyword*. *Keyword* adalah kata-kata yang menjelaskan objek-objek dalam suatu citra.
- Jumlah citra yang tersimpan dalam database sejumlah 100 citra.
- Terdapat 15-20 *word* dalam kamus kata.

#### 1.5 Metodologi Pembuatan

Metodologi pembuatan perangkat lunak yang digunakan dalam Tugas akhir ini sebagai berikut :

a. Studi Literatur

Pada tahap ini dipelajari sejumlah literatur mengenai konsep dan teknologi yang digunakan. Literatur yang digunakan meliputi buku, paper referensi, dan dokumentasi internet.

b. Perancangan Perangkat Lunak

Pada tahap ini dirancang sistem yang akan dibuat. Sistem tersebut akan menjadi acuan dalam tahap Pembuatan Perangkat Lunak.



c. Implementasi Perangkat Lunak

Dalam tahap ini dilakukan implementasi berdasarkan rancangan yang telah dibuat pada tahap sebelumnya.

d. Uji Coba dan Evaluasi

Pada tahap ini dilakukan uji coba program, mengevaluasi jalannya dan efisiensi program, dan mengadakan perbaikan jika ada kekurangan.

e. Penyusunan Buku Tugas Akhir

Penyusunan buku Tugas Akhir dilakukan untuk memberikan penjelasan-penjelasan berkenaan dengan aplikasi yang telah dibuat dan juga sebagai dokumentasi dari pelaksanaan tugas akhir

## 1.6 Sistematika Penulisan

Uraian sistematika penulisan pada Tugas Akhir ini bertujuan agar perancangan dan pembuatan perangkat lunak yang dibahas menjadi mudah dipahami, jelas dan sistematis untuk tiap-tiap bab atau sub bahasan. Bab 1 berisi latar belakang, tujuan, batasan masalah serta permasalahan yang dihadapi dalam pengerjaan tugas akhir ini. Bab 2 membahas teori segmentasi *Mean-Shift* dalam proses awal CBIR. Bab 3 membahas tentang proses CBIR dengan tiga tahapan besar yaitu pelatihan, pengenalan objek dan pencarian data citra dengan menggunakan algoritma *Expectation Maximization*, *K-Means*, dan *Nearest-neighbor*. Bab 4 Perancangan Perangkat Lunak menjelaskan tentang perancangan proses, dan perancangan algoritma. Bab 5 membahas tentang implementasi. Bab 6 menguraikan tentang lingkungan uji coba, dan pelaksanaan uji coba pada aplikasi

yang telah dibuat. Bab 7 berisi tentang kesimpulan dan saran yang dapat digunakan untuk proses pengembangan lebih lanjut dari aplikasi yang telah dibuat.



betapa berat rasa di hati ketika  
kami menyaksikan bencana yang mencabik cabik bangsa ini  
sementara kita menyerah pada kehinaan dan  
pasrah oleh keputusan  
kami ingin agar bangsa ini mengetahui  
bahwa kami membawa misi yang bersih dan suci  
bersih dari ambisi pribadi  
bersih dari kepentingan dunia dan bersih dari hawa nafsu  
kami tidak mengharap sesuatuapun dari manusia  
kami tidak mengharap harta benda atau imbalan lainnya  
tidak juga popularitas  
apalagi sekadar ucapan terimakasih  
yang kami harap adalah terbentuknya  
indonesia yang lebih baik dan bermartabat  
serta kebaikan dari ALLAH SWT

**IDEALISME KAMI-2 PPSDMS**

## **BAB II**

### **SEGMENTASI MEAN SHIFT**



## BAB 2

### Segmentasi *Mean Shift*

Penggunaan algoritme *mean shift* sebagai algoritme segmentasi dari *gradien* sebuah fungsi kerapatan dan prosedur iterasi telah dikembangkan oleh Fukunaga dan Hostetler [3]. Dengan menerapkan *mean shift* yang mengkombinasikan proses segmentasi dan *filtering* dalam domain spasial akan menghasilkan citra dengan kualitas tinggi. Metode ini secara konsep sangat sederhana yang berdasar pada ide iterasi penggeseran sebuah jendela berukuran tertentu berdasarkan nilai *mean*/rata-rata pada data.

Subbab 2.1 menjelaskan tentang dasar teori *mean shift*, perkiraan gradien kerapatan dan definisi vektor *mean shift*. Subbab 2.2 menjelaskan penerapan algoritme *mean shift* dalam tahapan segmentasi citra, sebagai tahapan awal/*pre-processing* dari proses pembuatan Sistem Temu Kembali Berbasis Isi.

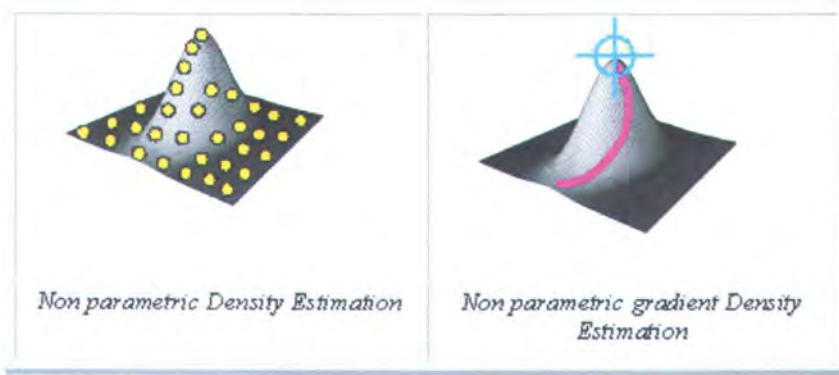
#### 2.1 Dasar Teori Mean Shift

*Mean Shift* adalah cara dalam mencari model dari sekumpulan data contoh, berdasarkan fungsi perkiraan kerapatan dalam  $R^d$  (Ruang dengan d-dimensi).

Analisis perkiraan kerapatan dibagi menjadi dua bagian :

1. *Non parametric Density Estimation*, perkiraan kerapatan berdasarkan keseluruhan data diskrit.
2. *Non parametric Gradient Density Estimation*, perkiraan kerapatan berdasarkan pergerakan *gradient* kerapatan.





Gambar 2.3 Density Estimation

### 2.1.1 Multivariate kernel density

Semisal himpunan  $\{x_i\}_{i=1,\dots,n}$  adalah kumpulan dari  $n$ -titik dalam dimensi  $-d$  Euclidian  $R^d$ . Perkiraan **Multivariate kernel density** [6] didapatkan dari kernel [7]  $K(x)$  dan jendela dengan jari-jari  $h$  yang dihitung pada titik  $x$  yang didefinisikan dengan

$$f^*(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (2.1)$$

Kernel yang menghasilkan *minimum mean integrated square error* (MISE) [3] bernilai error minimum, adalah *kernel Epachnikov*  $K_E$ .

$$K_E(x) = \begin{cases} \frac{1}{2} c_d^{-1} (d+2) (1 - x^T x), & \text{Jika } x^T x < 1 \\ 0 & \text{Jika } x^T x \geq 1 \end{cases} \quad (2.2)$$

dengan  $C_d$  adalah volume dari unit bidang dimensional  $-d$ .

### 2.1.3 Diferensiasi Kernel

Penurunan kernel mendefinisikan gradien kerapatan sebagai gradien dari perkiraan *kernel density* (2.1)

$$\nabla^* f(x) = \nabla f^*(x) = \frac{1}{nh^d} \sum_{i=1}^n \nabla K\left(\frac{x-x_i}{h}\right) \quad (2.3)$$

Kondisi pada *kernel*  $K(x)$  dan jari-jari jendela  $h$  menjamin konsistensi rata-rata kuadrat (*mean square*), dan konsistensi uniform yang dijelaskan pada (2.6).

Fungsi  $f^*(x)$  berada pada fungsi *continue*, maka agar **fungsi perkiraan *kernel density* (2.1)** menjadi fungsi diskrit, maka fungsi diturunkan terhadap  $x$

$$\begin{aligned} \nabla^* f(x) &= \nabla f^*(x) = \frac{1}{nh^d} \sum_{i=1}^n \nabla K\left(\frac{x-x_i}{h}\right) \\ &= \frac{1}{nh^d} \sum_{i=1}^n \nabla \left[ \frac{d+2}{2(c_d)} \left(1 - \left(\frac{x-x_i}{h}\right)^T \left(\frac{x-x_i}{h}\right)\right) \right] \\ &= \frac{1}{nh^d} \frac{d+2}{2(c_d)} \sum_{i=1}^n \nabla \left[ \left(\frac{h^2}{h^2} - \left(\frac{x-x_i}{h}\right)^T \left(\frac{x-x_i}{h}\right)\right) \right] \\ &= \frac{1}{nh^d} \frac{d+2}{2h^2(c_d)} \sum_{i=1}^n (h^2 - (x-x_i)^T (x-x_i)) \\ &= \frac{1}{nh^d} \frac{d+2}{2h^2(c_d)} \sum_{i=1}^n \nabla [h^2 - (x-x_i)^2] && \text{diturunkan terhadap } x \\ &= \frac{1}{nh^d} \frac{d+2}{2h^2(c_d)} \sum_{i=1}^n 2(x-x_i) \\ &= \frac{1}{nh^d} \frac{2(d+2)}{2h^2(c_d)} \sum_{i=1}^n (x-x_i) \end{aligned}$$

#### 2.1.4 Diferensiasi *Kernel Epanichnikov*

Untuk *Kernel Epanichnikov* (2.2), maka penurunan perkiraan *gradient density* (2.3) menjadi



$$\begin{aligned}
\bar{\nabla}^* f(x) &= \frac{1}{nh^d c_d} \frac{d+2}{h^2} \sum_{x_i \in S_h^*(x_k)} [x_i - x] \\
&= \left[ \frac{n_x}{n(h^d c_d)} \right] \frac{d+2}{h^2} \left[ \frac{1}{n_x} \sum_{x_i \in S_h^*(x_k)} [x_i - x] \right]
\end{aligned} \tag{2.4}$$

Region  $S_h(x)$  adalah sebuah *hypersphere*/ *jendela* dengan **jari-jari  $h$** , yang memiliki **volume  $h^d c_d$**  berpusat di  $x$ , dan mempunyai  $n_x$  titik data. Tahap terakhir pada persamaan (2.4) :

$$M_h(x) = \frac{1}{n_x} \sum_{x_i \in S_h^*(x_k)} [x_i - x] = \frac{1}{n_x} \sum_{x_i \in S_h^*(x_k)} x_i - x \tag{2.5}$$

Persamaan 2.5 tersebut disebut ***mean shift*** sederhana.

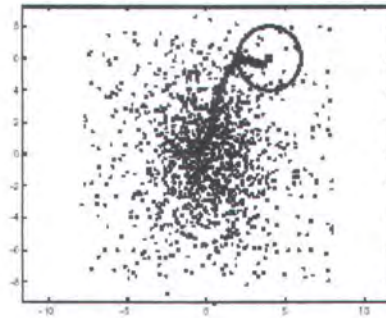
Kuantitas  $\left[ \frac{n_x}{n(h^d c_d)} \right]$  adalah perkiraan *kernel density*  $\bar{f}^*(x)$  yang dihitung dengan *hypersphere*  $S_h(x)$  (*kernel uniform*) sehingga rumus (2.4) dapat ditulis menjadi  $\bar{\nabla}^* f(x) = \bar{f}^*(x) \frac{d+2}{h^2} M_x(x)$

$$\text{yang menghasilkan Vektor Mean Shift } M_x(x) = \frac{h^2}{d+2} \frac{\bar{\nabla}^* f(x)}{\bar{f}^*(x)} \tag{2.7}$$

Persamaan (2.7) adalah turunan pertama dari persamaan (2.6) dan menunjukkan bahwa estimasi dari gradien normal dapat diperoleh dengan menghitung *sample mean shift* dalam *kernel uniform* yang berpusat di  $x$ . Vektor *mean shift* mempunyai arah gradien dari estimasi kerapatan pada  $x$ , jika estimasi diperoleh dengan *Epanechnikov kernel*.

Karena vektor *mean shift* selalu mengacu pada arah dari peningkatan maximum dalam kerapatan, maka dapat ditentukan jalur (*path*) ke maksimal

kerapatan lokal (*local maximum density*). Gambar 2.1 menunjukkan contoh model kerapatan dengan perpindahan vektor *mean shift*.



**Gambar 2.4** Perpindahan vektor Mean Shift ke arah kerapatan maksimal lokal dari perhitungan mean shift

Perumusan algoritme *Mean Shift* didapatkan dari tahapan sebagai berikut :

1. Menghitung vektor *mean Shift*  $M_h(x)$  .
2. Perpindahan jendela  $S_h(x)$  dengan melihat  $M_h(x)$  yang menuju konvergensi .

### 2.1.5 Konvergensi *Mean Shift*

Semisal himpunan  $\{y_{k=1}\}_{k=1,2,\dots}$  adalah lokasi prosedur *Mean Shift* , sehingga untuk setiap  $k= 1, 2, \dots$

$$y_{k+1} = \frac{1}{n_k} \sum_{x_i \in S_h(y_k)} x_{i,j} \quad (2.8)$$

$y_1$  adalah pusat jendela awal dengan  $n_k$  adalah jumlah titik data pada jendela  $S_h(y_k)$  yang berpusat pada  $y_k$  .

Konvergensi *Mean Shift* telah dijelaskan sebagai hasil dari persamaan (2.7) [9] . Meskipun telah terbukti bahwa vektor *mean shift*  $M_h(x)$  didapatkan dari arah perkiraan kerapatan pada titik  $x$ , namun belum menjelaskan bahwa perkiraan



kerapatan pada lokasi  $\{y_{k=1}\}_{k=1,2,\dots}$  akan naik secara monoton. Perpindahan arah dari gradien menjamin *hill climbing* untuk tahapan *infinitesimal*. Teorema konvergensi *mean shift* untuk data diskrit dijelaskan pada teorema 1.

**Teorema 1:**

Jika semisal  $f^*_E(k) = \{f^*_E(y_k, K_E)\}$  adalah rangkaian dari perkiraan kerapatan yang dihasilkan dari penggunaan Kernel Epanichnikov pada titik  $\{y_k\}_{k=1,2,\dots}$  yang merupakan lokasi prosedur *mean shift* dengan *uniform kernel*, maka rangkaian tersebut akan konvergen.

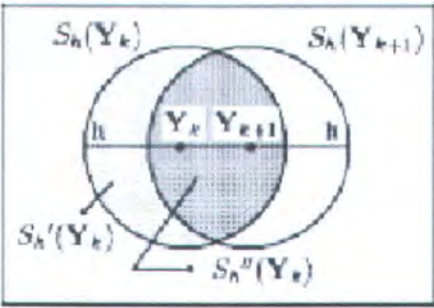
**Pembuktian:**

Karena himpunan  $\{x_i\}_{i=1,\dots,n}$  adalah titik yang dibatasi dengan nilai  $n$ , maka urutan  $f^*_E$  juga terbatas.  $f^*_E$  akan naik secara monoton, jika  $y_k \neq y_{k+1}$  maka  $f^*_E(k) < f^*_E(k+1)$  untuk semua  $k=1,2, \dots$

Semisal  $n_k$ ,  $n'_k$ , dan  $n''_k$  dengan  $n_k = n'_k + n''_k$  adalah jumlah titik data yang ada pada jendela berdimensi  $d$ .  $S_h(y_k), S'_h(y_k) = S_h(y_k) - S''_h(y_k)$  dan  $S''_h(y_k) = S_h(y_k) \cap S_h(y_{k+1})$ .

Dengan bersasumsi bahwa letak awal jendela pada di titik  $y_k$  dan menggunakan definisi dari perkiraan kerapatan gradien (2.1), Kernel Epanichnikov (2.2) dan mengabaikan bahwa  $\|y_k - x_i\|^2 = \|x_i\|^2$  maka diperoleh

$$\begin{aligned} f^*_{\scriptscriptstyle E}(k) &= f^*_k(Y_k, K_E) = \frac{1}{nh^d} \sum_{x_i \in S_h(y_k)} K_E\left(\frac{y_{k+1} - x_i}{h}\right) \\ &= \frac{1}{nh^d} \sum_{x_i \in S_h(y_k)} \frac{d+2}{2(c_d)} \left(1 - \left(\frac{y_k - x_i}{h}\right)^T \left(\frac{y_k - x_i}{h}\right)\right) \\ &= \frac{1}{nh^d} \frac{d+2}{2(c_d)} \sum_{x_i \in S_h(y_k)} \left(1 - \frac{\|y_k - x_i\|^2}{h^2}\right) \\ &\Leftrightarrow \|y_k - x_i\|^2 = \|x_i\|^2 \\ &= \frac{d+2}{2n(h^d c_d)} \sum_{x_i \in S_h(y_k)} \left(1 - \frac{\|x_i\|^2}{h^2}\right) \end{aligned} \tag{2.9}$$



Gambar 2.5 Jendela *Mean Shift*

Pada gambar 2.3 dijelaskan jendela dimensi ke- $d$  yang digunakan untuk membuktikan konvergensi dari  $S_h(y_k)$ ,  $S'_h(y_k)$ , dan  $S''_h(y_k)$ . Titik  $y_{k+1}$  adalah mean dari titik dalam  $S_h(y_k)$ . Terdapat dua buah himpunan  $S_h(y_k)$  dan  $S_h(y_{k+1})$  yang beririsan pada  $S''_h(y_{k+1})$ .  $S'_h(y_k) = S_h(y_k) - S''_h(y_k)$  dengan jumlah titik pada  $S_h(Y_k) = n_k$ ,  $n_k = n''_k + n'_k$ .



Ketika  $K_E$  juga bernilai *non negative*/ tidak negatif maka

$$\begin{aligned}
 f^*_E(k+1) &= f^*_{k+1}(Y_{k+1}, K_E) \geq \frac{1}{nh^d} \sum_{x_i \in S'_h(y_k)} K_E\left(\frac{y_{k+1} - x_i}{h}\right) \\
 &= \frac{1}{nh^d} \sum_{x_i \in S'_h(y_k)} \frac{d+2}{2(c^d)} \left(1 - \left(\frac{y_{k+1} - x_i}{h}\right)^T \left(\frac{y_{k+1} - x_i}{h}\right)\right) \\
 &= \frac{d+2}{2n(h^d c_d)} \sum_{x_i \in S'_h(y_k)} \left(1 - \left(\frac{\|y_{k+1} - x_i\|^2}{h^2}\right)\right)
 \end{aligned} \tag{2.10}$$

Dengan mengingat bahwa  $n'_k = n_k - n''_k$ , maka diperoleh

$$\begin{aligned}
 f^*_E(k+1) &= f^*_{k+1}(Y_{k+1}, K_E) \geq \frac{1}{nh^d} \sum_{x_i \in S'_h(y_k)} K_E\left(\frac{y_{k+1} - x_i}{h}\right) \\
 &= \frac{d+2}{2n(h^d c_d)} \sum_{x_i \in S'_h(y_k)} \left(1 - \left(\frac{\|y_{k+1} - x_i\|^2}{h^2}\right)\right) \\
 &= \frac{d+2}{2n(h^d c_d)h^2} \sum_{x_i \in S'_h(y_k)} (h^2 - \|y_{k+1} - x_i\|^2) \\
 &= \frac{d+2}{2n(h^d c_d)h^2} \left\{ \sum_{x_i \in S'_h(y_k)} (h^2) - \sum_{x_i \in S'_h(y_k)} \|y_{k+1} - x_i\|^2 \right\} \\
 &= \frac{d+2}{2n(h^d c_d)h^2} \left\{ n''_k h^2 - \sum_{x_i \in S'_h(y_k)} \|y_{k+1} - x_i\|^2 \right\}
 \end{aligned}$$

karena  $S'_h(y_k) = S_h(y_k) - S''_h(y_k)$  dan  $n'_k = n_k - n''_k$

maka  $n'_k h^2 = n_k h^2 - n''_k h^2 \rightarrow n''_k h^2 = n_k h^2 - n'_k h^2 \rightarrow n''_k h^2 = \sum_{x_i \in S_h(y_k)} \|x_i\|^2 - n'_k h^2$

$$\begin{aligned}
 &= \frac{d+2}{2n(h^d c_d)} \left\{ \sum_{x_i \in S_h(y_k)} \|x_i\|^2 - n''_k h^2 - \sum_{x_i \in S'_h(y_k)} \|y_{k+1} - x_i\|^2 \right\} \\
 &= \frac{d+2}{2n(h^d c_d)} \left\{ \sum_{x_i \in S_h(y_k)} \|x_i\|^2 - \sum_{x_i \in S_h(y_k)} \|y_{k+1} - x_i\|^2 - n''_k h^2 \right\}
 \end{aligned} \tag{2.11}$$

dengan definisi  $\|y_{k+1} - x_i\|^2 \geq h^2$  untuk  $S_h^n(y_k)$ , maka dapat disederhanakan menjadi

$$\sum_{x_i \in S_h^n(y_k)} \|y_{k+1} - x_i\|^2 \geq n'_k h^2 \quad (2.12)$$

Dengan menggunakan rumus (2.11), (2.12) dan (2.8) diperoleh

$$f^*_E(k+1) - f^*_{k+1}(Y_{k+1}, K_E) \geq \frac{d+2}{2n(h^d c_d)h^2} \left[ \sum_{x_i \in S_h(y_k)} \|x_i\|^2 - \sum_{x_i \in S_h(y_k)} \|y_{k+1} - x_i\|^2 \right]$$

Bentuk  $\sum_{x_i \in S_h(y_k)} \|x_i\|^2 - \sum_{x_i \in S_h(y_k)} \|y_{k+1} - x_i\|^2$  serupa dengan  $(x-y)(x+y) = x^2 - y^2$

$$\text{dengan } x^2 = \|x_i\|^2, y^2 = \|y_{k+1} - x_i\|^2$$

$$\text{maka } \|x_i\|^2 - \|y_{k+1} - x_i\|^2 = (x_i - y_{k+1} + x_i)(x_i + y_{k+1} - x_i)$$

$$= (2x_i - y_{k+1})(y_{k+1})$$

$$= (2x_i y_{k+1} - y_{k+1}^2)$$

Sehingga

$$f^*_E(k+1) - f^*_{k+1}(Y_{k+1}, K_E) \geq \frac{d+2}{2n(h^d c_d)h^2} \left[ \sum_{x_i \in S_h(y_k)} \|x_i\|^2 - \sum_{x_i \in S_h(y_k)} \|y_{k+1} - x_i\|^2 \right]$$

$$= \frac{d+2}{2n(h^d c_d)h^2} \left[ \sum_{x_i \in S_h(y_k)} (2x_i y_{k+1} - y_{k+1}^2) \right]$$

$$= \frac{d+2}{2n(h^d c_d)h^2} \left[ \sum_{x_i \in S_h(y_k)} 2x_i y_{k+1} - \sum_{x_i \in S_h(y_k)} y_{k+1}^2 \right]$$

$$= \frac{d+2}{2n(h^d c_d)h^2} \left[ 2y_{k+1} \sum_{x_i \in S_h(y_k)} x_i - \sum_{x_i \in S_h(y_k)} y_{k+1}^2 \right]$$



$$\begin{aligned}
&= \frac{d+2}{2n(h^d c_d)h^2} \left[ 2y_{k+1} \sum_{x_i \in S_h(y_k)} x_i - y_{k+1}^2 \sum_{x_i \in S_h(y_k)} 1 \right] \\
&= \frac{d+2}{2n(h^d c_d)h^2} \left[ 2y_{k+1} \sum_{x_i \in S_h(y_k)} x_i - n_k y_{k+1}^2 \right] \\
&= \frac{d+2}{2n(h^d c_d)h^2} \left[ 2y_{k+1}^T \sum_{x_i \in S_h(y_k)} x_i - n_k \|y_{k+1}\|^2 \right] \\
&= \frac{d+2}{2n(h^d c_d)h^2} n_k \|y_{k+1}\|^2 \tag{2.13}
\end{aligned}$$

yang akan bernilai positif kecuali ketika  $Y_k = Y_{k+1} = 0$

Ketika dibatasi dan naik secara monoton, maka  $f^*$  menjadi konvergen.

Dengan catatan jika  $Y_k = Y_{k+1} = 0$ ,  $Y_k$  adalah limit dari  $f^*$ ,  $Y_k$  adalah titik yang tetap dari prosedur *mean shift*.

## 2.2 Penerapan Algoritma *Mean Shift* pada Segmentasi Citra

Sebuah citra umumnya memiliki 2 dimensi dari  $r$ -vektor dimensi/piksel,  $r$  bernilai 1 jika citra *gray level*, dan  $r$  bernilai 3 untuk citra berwarna. Domain lapisan ini dinamakan *spatial domain*. Informasi *gray level* dan warna adalah mewakili *range domain*. Setelah dilakukan normalisasi dengan parameter global  $\{\sigma_s, \sigma_r\}$  kedalam *spatial domain* dan *range domain*, maka lokasi dan vektor jarak dapat dihitung untuk mendapatkan sebuah domain *spatial-range* dari dimensi  $d=r+2$ . Hasil dari *mean shift filter* dari sebuah citra didefinisikan sebagai informasi jarak yang diperoleh dari titik konvergen.

### 2.2.1 Mean Shift Filter

Semisal himpunan  $\{x_i\}_{i=1,\dots,n}$  adalah himpunan n-titik data dari d-dimensi citra awal, dan himpunan  $\{z_j\}_{j=1,\dots,n}$  adalah himpunan titik-titik konvergen titik filter pada *spatial-range domain*. Notasi s dan r adalah bagian *spatial-range* dari vektor-vektor.

Pada setiap iterasi k nilai mean dilakukan perhitungan vektor mean shift sebagai berikut:

untuk setiap  $j=1, \dots, n$

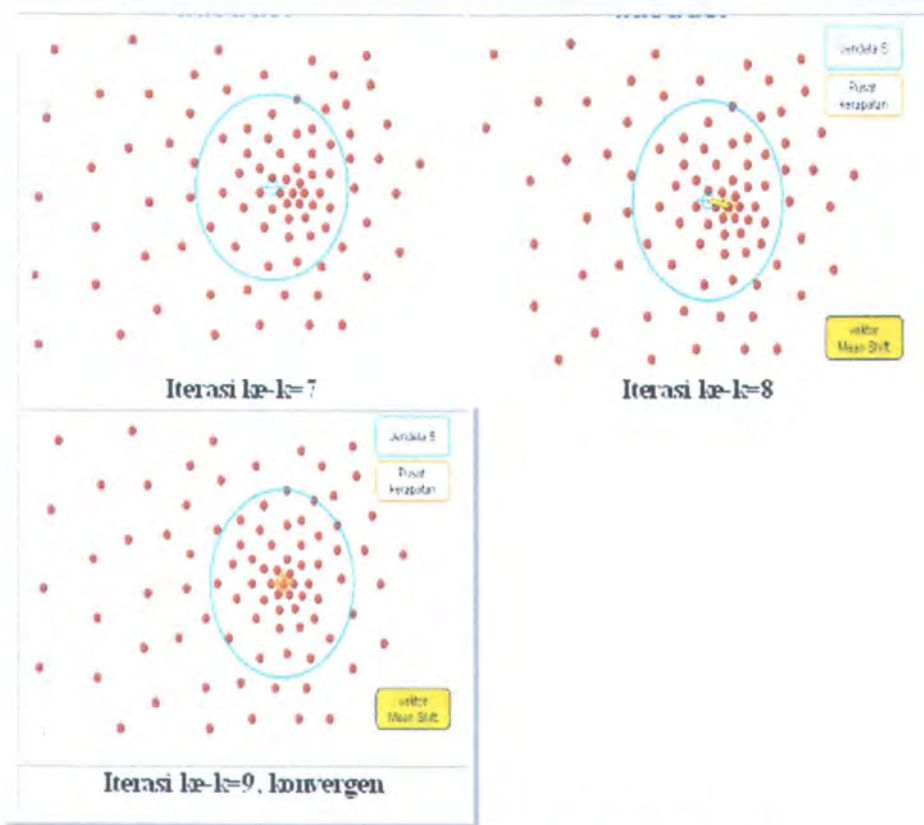
1. Inisialisasi  $k=1$  dan  $y_k = x_j$
2. Hitung  $y_{k+1} = \frac{1}{n_k} \sum_{x_i \in S_1(y_k)} x_i, k \leftarrow k + 1$  sampai mencapai konvergen
3. Memberi nilai  $z_j = (x_j^s, y_{conv}^r)$

Langkah terakhir menunjukkan data yang telah difilter pada *spatial domain* dari  $x_j$  akan mempunyai titik konvergen  $y_{conv}$ . Jumlah dari titik pada jendela  $S_1(y_k)$  dari jari-jari 1 dan yang berpusat pada  $y_k$  adalah  $n_k$ . Jari-jari dari jendela digunakan dalam tahapan normalisasi.





Gambar 2.6 Ilustrasi Iterasi *Mean Shift* Iterasi 1-6



Gambar 2.7 Ilustrasi Iterasi *Mean Shift* Iterasi 7-9

### 2.2.3 Segmentasi *Mean Shift*

Data diasumsikan dinormalisasi dengan  $\{\sigma_s, \sigma_r\}$  dan himpunan  $\{x_i\}_{i=1, \dots, n}$  adalah kumpulan dari  $n$ -titik data, serta himpunan  $\{z_j\}_{j=1, \dots, n}$  adalah titik-titik konvergen dan himpunan  $\{L_j\}_{j=1, \dots, n}$  adalah sekumpulan label dalam nilai skalar. Maka langkah-langkah algoritme segmentasi *mean shift* adalah sebagai berikut:

1. Untuk setiap  $j=1, \dots, n$  hitung proses *mean shift filter* untuk  $x_j$  dan menyimpan titik yang konvergen pada  $z_j$ .
2. Mengidentifikasi *cluster*  $\{C_p\}_{p=1, \dots, m}$  dari titik konvergen yang terhubung dengan semua  $z_j$  yang lebih dekat 1 dari domain yang lain.



3. Untuk setiap  $j=1, \dots, n$ , diberi label dengan  $L_j = \{p \mid z_j \in C_p\}$
4. Menghilangkan *cluster* yang lebih kecil dari  $M$  piksel, nilai  $M$  telah ditentukan sebelumnya.

Langkah pertama dari segmentasi mean shift adalah proses *mean shift filter*. Semua informasi tentang titik konvergen d-dimensi disimpan pada  $z_j$ . Jumlah *cluster*  $m$  ditentukan oleh nilai  $\{\sigma_s, \sigma_r\}$ .

Citra pada data set disegmentasi menggunakan algoritma *Mean Shift* [2] sehingga menjadi beberapa *region*. Setiap citra data set rata-rata mempunyai 5-10 *region*. Langkah Pertama yang dilakukan adalah mengubah data citra dalam domain RGB ke domain fitur. Untuk memastikan *isotropy* dari domain fitur maka perlu digunakan pengukuran dengan *Euclidian distances*. Oleh karena itu, domain  $L^*u^*v$  digunakan.  $L^*u^*v$  memiliki koordinat yang berkaitan dengan nilai RGB dalam transformasi nonlinear. Informasi kromatik dibawa oleh  $u^*$  dan  $v^*$ , dan  $L^*$  membawa informasi kecerahan relatif citra hasil segmentasi citra tersebut berupa *region-region*. Setiap *region* memiliki ciri-ciri tertentu yang akan memberikan informasikan sebuah objek dalam citra.



berjalan seorang pria muda  
dengan jaket lusuh di pundaknya  
di sela bibir mengering  
terselip sebatang ruput liar

jelas menatap awan berarak  
wajah murung semakin terlihat  
dengan langkah gontai tak terarah  
keringat bercampur debu jalanan

engkau sarjana muda  
resah mencari kerja  
mengandalkan ijasahmu  
5 tahun lamanya bergelut dengan buku  
'tuk jaminan masa depan  
langkah kakimu terhenti  
di depan halaman sebuah jawatan

(setengah putus asa dia berucap) "maaf ibu ..."

**IWAN FALS- SARJANA MUDA**

### BAB III

## SISTEM TEMU KEMBALI CITRA BERBASIS ISI



## BAB 3

### Sistem Temu Kembali Citra Berbasis Isi

Tiga tahapan Sistem Temu Kembali Citra berbasis isi yaitu pelatihan data citra, pengenalan data citra baru, dan pencarian data citra. Subbab 3.1 menjelaskan tahapan pelatihan data, sedangkan Subbab 3.2 menjelaskan tahapan pengenalan data citra baru. Tahapan pencarian data dijelaskan pada subbab 3.3.

#### 3.1 Pelatihan Data Citra

Pada saat pelatihan data, diperoleh dua macam himpunan untuk setiap citra, yaitu himpunan *keyword* dan himpunan *blob*. *Keyword* adalah *word* yang mewakili objek-objek yang secara manual diberikan pada data citra. Untuk mendapatkan *blob* (hasil pengelompokkan dari *region*), setiap data citra disegmentasi dengan algoritme *mean shift* untuk mendapatkan *region-region*. Pada setiap *region* dilakukan proses ekstraksi fitur dasar untuk menghasilkan vektor fitur. Kemudian, vektor fitur dari setiap *region* pada semua data citra dari hasil pelatihan data dikelompokkan menjadi *blob-blob* dengan menggunakan algoritme *K-Means*.

Dengan demikian, untuk setiap citra didapatkan data mengenai *blob-blob* dalam citra tersebut sekaligus *keyword* (atau jenis objek) yang mewakili *blob* tersebut. Untuk menyusun tabel probabilitas mengenai kemungkinan bahwa suatu *blob* adalah suatu objek tertentu (atau *keyword* tertentu) digunakan algoritme



*Expectation Maximization*, karena ada data yang 'missing'(tersembunyi) yaitu korespondensi eksplisit yang menghubungkan antara *blob* dan *word*. Tabel probabilitas yang dihasilkan selanjutnya digunakan untuk memprediksi (mengenali) objek-objek apa saja yang ada di dalam citra.

### 3.1.1 Segmentasi Citra

Untuk mendapatkan blob-blob dari data visual maka Data Citra perlu disegmentasi dengan menggunakan algoritme *Mean Shift* seperti telah dijelaskan pada bab 2.

### 3.1.2 Ekstraksi Fitur Dasar Warna

Informasi yang ada dalam *region-region* dari setiap data citra digunakan untuk menentukan ciri-ciri dari objek dalam citra . Oleh karena itu, *region-region* tersebut diekstraksi dan ditransformasikan dalam vektor fitur. Proses ekstraksi bertujuan untuk mendapatkan vektor fitur berupa fitur dasar dari tiap-tiap region hasil segmentasi.

#### 3.1.2.1 Fitur Dasar Centroid/pusat massa, Area-percontour

Pusat massa :

$$\bar{x} = \frac{\sum_{x=0}^m \sum_{y=0}^n xB(x, y)}{A}, \bar{y} = \frac{\sum_{x=0}^m \sum_{y=0}^n yB(x, y)}{A} \quad (3.1)$$

Area adalah jumlah pixel dalam region :

$$A = \sum_{x=0}^m \sum_{y=0}^n B(x, y) \quad (3.2)$$

Perimeter adalah panjang dari tepi citra /jumlah hpixel yang mengelilingi citra

Luas region dibagi akar dari panjang contour region :

$$A_{contour} = A / P$$

Mean dan varian yang digunakan berdasarkan nilai intensitas warna dalam domain  $L^*u^*v$ ,  $L^*a^*b$ , RGB, dan CrCb dijelaskan pada subbab 3.1.2.2.

### 3.1.2.2 Fitur *Mean* dan *Varian*

Warna adalah faktor penting dalam pembentukan persepsi manusia terhadap pengidentifikasian objek dan materi. Persepsi terhadap warna tergantung pada kedua hal yaitu fisik cahaya dan proses kompleks dari mata dan otak yang mengintegrasikan sifat dari rangsangan.

Dengan adanya perbandingan dan pemrosesan warna, informasi warna dapat digunakan secara efektif untuk citra dan visi komputer. Warna mempunyai banyak informasi per piksel, sering menyediakan klasifikasi yang kompleks.

Histogram warna paling sering digunakan dalam pengolahan citra dan telah terbukti sangat bermanfaat. Namun demikian karakterisasi secara keseluruhan masih kurang, contohnya untuk membedakan kebun bunga warna oranye dan seekor harimau, karena kurangnya informasi mengenai bagaimana warna didistribusikan secara parsial dan kurangnya informasi lain yang bisa membedakan kedua objek tersebut



Empat format warna yang diekstraksi dari setiap *region-region* citra adalah *mean* dan varian dari domain warna CrCb, RGB, L\*a\*b, L\*u\*v.

*Mean* adalah rata rata dari sekumpulan nilai pada piksel citra :

$$\alpha = \frac{1}{N} \sum_{x=0}^m \sum_{y=0}^n B(x, y) \quad (3.3)$$

*Varian* adalah mengukur penyebaran nilai pada data citra *region* :

$$\sigma = \sqrt{\frac{1}{N} \sum_{x=0}^m \sum_{y=0}^n (B(x, y) - \alpha)^2} \quad (3.4)$$

### 3.1.2.2.1 Domain Warna RGB

Pengkodean Warna Trikromatik RGB (*Red-Green-Blue*) dalam grafik biasanya menggunakan 3 *bytes* yang berarti menyediakan  $(2^8)^3$  atau 16 juta warna, lebih tepatnya 16 juta kode, karena manusia tidak dapat menangkap sejumlah besar warna itu, tetapi komputer dapat melakukannya. Pengkodean 24 bit menggunakan 8 bit untuk warna merah, hijau dan biru.

Untuk tujuan interpretasi dari manusia dan komputer, maka digunakan normalisasi data citra dan mentransformasi ke domain warna lain

$$\text{Intensity } I = (R + G + B)/3$$

$$\text{Normalized red} = R/(R + G + B)$$

$$\text{Normalized green} = G/(R + G + B)$$

$$\text{Normalized blue} = B/(R + G + B)$$

Dengan penskalaan nilai antara 0 dan 1

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} Xr & Xg & Xb \\ Yr & Yg & Yb \\ Zr & Zg & Zb \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.431 & 0.342 & 0.178 \\ 0.222 & 0.707 & 0.071 \\ 0.020 & 0.130 & 0.939 \end{bmatrix} \times \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.5)$$

### 3.1.2.2.2 Domain CrCb

Model ini adalah standard untuk pengkodean digital gambar TV pada 525 dan 625 baris tingkatan. CrCb tidak mempunyai koordinat kromatik, tidak mempunyai matrik XYZ. Model ini hanya sebagai representasi digital dari sinyal RGB pada bentuk Y'CbCr.

$$\begin{bmatrix} Y' \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + A \times \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad (3.6)$$

Menggunakan jenis konversi Rec. 601 (SDTV)

$$\begin{bmatrix} 0.25878824 & 0.50412941 & 0.09790588 \\ -0.1482229 & -0.29099279 & 0.43921589 \\ 0.43921589 & -0.36778831 & -0.07142737 \end{bmatrix}$$

### 3.1.2.2.3 Domain CIE L\*u\*v

L\*u\*v di gunakan pada produk beberapa produk video digital dan algoritme kompresi seperti JPEG dan MPEG. L\*u\*v memiliki potensi yang baik untuk kompresi citra digital dan video daripada sistem warna yang lain.

$$L = 116 \times ((Y / Y_n)^{\frac{1}{3}}) \text{ untuk } (Y / Y_n) > 0.008856$$

$$L = 903.3 \times (Y / Y_n) \text{ untuk } (Y / Y_n) \leq 0.008856$$

$$\begin{aligned} u &= 13 \times L \times (u' - u'_n) \\ v &= 13 \times L \times (v' - v'_n) \end{aligned} \quad (3.7)$$

Skala  $L^*$  mulai dari 0 -100 untuk nilai  $Y/Y_n$  antara 0-1

$u'$  dan  $v'$  ditentukan melalui persamaan di bawah ini :

$$\begin{aligned} u' &= 4X(X + 15Y + 3Z) = 4x(-2x + 12y + 3) \\ v' &= 9X(X + 15Y + 3Z) = 9x(-2x + 12y + 3) \end{aligned}$$

### 3.1.2.2.3 Domain CIE $L^*a^*b$

Model  $L^*a^*b$  color digunakan berdasarkan model yang diajukan oleh Commission Internationale d'Eclairage (CIE) tahun 1931 sebagai standard pengukuran warna internasional. Tahun 1976, model ini direvisi dan dinamakan CIE  $L^*a^*b$ .

Warna  $L^*a^*b$  didesain untuk menciptakan warna yang konsisten untuk berbagai jenis alat ( seperti monitor, printer, computer, atau scanner) digunakan untuk menciptakan atau output dari citra. Warna  $L^*a^*b$  terdiri dari *luminance* atau komponen kecerahan ( $L$ ) dan 2 komponen kromatik:

- Komponen a (dari hijau ke merah )
- Komponen b (dari biru ke kuning)

$$L^* = 116 \times (Y/Y_n)^{1/3} \text{ untuk nilai } Y/Y_n > 0.008856$$

$$L^* = 903.3 \times Y/Y_n \quad \text{Untuk nilai } Y/Y_n \leq 0.008856 \quad (3.8)$$

$$a^* = 500 \times (f(X/X_n) - f(Y/Y_n))$$



## ABSTRAK

*Sistem Temu Kembali Citra Berbasis Isi adalah sistem yang mengorganisir data citra hasil pengenalan objek, mengenali data citra baru dan mampu melakukan pencarian data citra berdasarkan objek dari citra dalam jumlah citra besar dan variasi gambar yang beragam. Salah satu implementasi penggunaan Sistem Temu Kembali Citra Berbasis Isi adalah pencarian koleksi barang di museum. User dapat memperoleh informasi gambar tentang barang-barang koleksi museum hanya dengan memasukkan nama objek dari barang koleksi.*

*Pada Tugas Akhir ini akan difokuskan pertama pada pembuatan Sistem Temu Kembali Citra Berbasis Isi dengan tiga tahapan. Pada tahapan pertama pelatihan data, data citra yang telah diberi label/keyword berisikan nama-nama objek pada citra disegmentasi menggunakan algoritme Mean Shift untuk menghasilkan region-region. Untuk menghasilkan vektor fitur citra maka dilakukan ekstraksi fitur dasar warna pada hasil segmentasi, mengelompokkan/clustering region ke dalam kelas-kelas/blob-blob dengan menggunakan algoritme K-Means berdasarkan vektor fitur, memprediksi keterhubungan antara blob dengan word dengan menggunakan algoritme Expectation Maximization. Tahap kedua, Pengenalan Data Citra baru dilakukan dengan menggunakan algoritme Nearest-Neighbour berdasarkan kedekatan vektor fitur data citra baru dengan data citra hasil pelatihan. Pada tahap ketiga dilakukan Pencarian Data Citra berdasarkan data input user berupa teks yang mewakili nama objek yang dicari dari data citra yang telah dihasilkan pada tahapan pelatihan data. Fokus kedua yaitu analisis kinerja Sistem Temu Kembali Citra Berbasis Isi hasil dari pemilihan fitur dasar warna dalam domain ( $L^*u^v$ , RGB,  $L^*a^b$ ,  $L^*a^b$  dan CrCb), penggunaan algoritme mean shift pada tahapan awal/preprocessing.*

*Berdasarkan hasil uji coba, penentuan jumlah klaster yang lebih banyak terbukti meningkatkan kinerja Sistem Temu Kembali Citra Berbasis Isi serta pemilihan fitur dasar warna domain  $L^*u^v$  lebih memiliki nilai kinerja yang lebih tinggi daripada domain warna yang lain ( $L^*u^v$ , RGB,  $L^*a^b$ ,  $L^*a^b$ , CrCb, dan kombinasinya)*

**Kata Kunci:** *Clustering, Expectation Maximization, K-Means, Mean Shift Nearest-Neighbour*

$b^*=200*(f(Y/Y_n)-f(Z/Z_n))$  dimana  $f(t)=t^{1/3}$  untuk  $t>0.008856$

$f(t)=7.787*t+16/116$  untuk  $t\leq 0.008856$

$L^*$  bernilai antara 0 dan 100.

### 3.1.3 Pengelompokkan vektor fitur dari setiap region dengan Algoritme K-Means

Dasar Teori K-Means dijelaskan pada subbab 3.1.3.1 dan penerapan clustering k-means pada Sistem Temu Kembali Berbasis Isi dijelaskan pada 3.1.3.2.

#### 3.1.3.1 Dasar Teori Algoritme K-Means

Algoritme *K-Means* adalah salah satu algoritme sederhana dari metode *unsupervised learning* [10] yang umum digunakan untuk menyelesaikan permasalahan *clustering*/pengelempokkan karena mudah untuk diimplementasikan. Tujuan utama dari *K-Means clustering* adalah membagi data sebanyak  $k$  kelompok, yang berpusat pada pusat/ *centroid*  $c_j$ ,  $j$  adalah kelompok data ke-1, ...,  $k$ . Cara yang digunakan adalah dengan menjauhkan sejauh mungkin antara satu pusat dengan pusat yang lain.

Langkah pertama adalah menempatkan setiap titik pada kelompok data dengan jarak terdekat dari pusat. Langkah pertama berhenti ketika keseluruhan titik telah ditempatkan pada  $k$ -kelompok data. Dari  $k$ -kelompok data tersebut didapatkan titik pusat  $k$  pertama. Langkah selanjutnya menempatkan setiap titik pada kelompok data berdasarkan jarak terpendek dari titik pusat  $k$  yang pertama. Langkah selanjutnya adalah dengan menghitung kembali titik  $k$  pusat yang kedua,

dengan mencari pusat dari tiap-tiap kelompok data. Iterasi berhenti ketika koordinat titik pusat k data tidak berubah.

Fungsi *K-Means* ;

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2 \quad (3.9)$$

$\|x_i^{(j)} - c_j\|$  : jarak antara titik  $x_i^{(j)}$  dengan titik pusat  $c_j$

Langkah algoritme *K-Means* :

1. Menentukan k-titik sebagai pusat kelompok data..
2. Menempatkan setiap titik data pada kelompok data ke-k yang mempunyai jarak terdekat antara titik data  $x_i^{(j)}$  dengan titik pusat  $c_j$ .
3. Ketika seluruh titik data telah ditempatkan pada k-kelompok, maka dihitung kembali letak titik pusat  $c_j$ .
4. Kembali ke langkah 2 dan 3, sehingga titik pusat tidak berubah letaknya.

Namun algoritme *K-Means* tidak dapat menunjukkan konfigurasi optimal pada pengelompokan data dengan k kelompok

### 3.1.3.2 Pengelompokan data Region dengan algoritme K Means

Pengelompokan data region bertujuan untuk mengelompokkan fitur-fitur berdasarkan ciri yang lekat pada tiap-tiap fitur. Pada gambar 3.1 terdapat beberapa citra yang memuat objek langit, *region-region* pada citra tersebut memiliki ciri-ciri tertentu yang menginformasikan kelas langit seperti luas *region*, rata-rata dan varian tingkat kecerahan warna.



*Region-region* yang bervektor fitur hampir sama dari semua *data set* dikelompokkan menjadi satu kelas yang disebut dengan *blob*. Pengelompokkan ini menggunakan algoritme *K-means*.

Gambar 3.1 Menunjukkan kemungkinan kesamaan dari ciri-ciri objek langit. Sehingga pada *region* langit pada ketiga gambar 3.1 mempunyai nilai vektor fitur dasar warna yang hampir sama.



Gambar 3.8 Kesamaan Citra dari Beberapa Citra



Gambar 3.9 *clustering region*

Gambar 3.2 menjelaskan citra hasil segmentasi memiliki empat *region* dan kemudian dikelompokkan berdasarkan ciri-ciri fitur dasar dari keseluruhan *region* citra yang ada pada database, sehingga diperoleh empat kelompok *blob* yaitu *b6*, *b5*, *b2*, dan *b4* pada setiap *region* dari citra tersebut.

3.1.4 Menghubungkan Blob dengan Word dengan Algoritme *Expectation Maximization*



Gambar 3.10 keterhubungan *blob* dengan *word*

Gambar 3.3.a menunjukkan citra dengan 4 *blob* (b15, b6, b6, b73) dan 2 *word* (burung, langit), jumlah *blob* lebih banyak daripada *word*. Gambar 3.3.b menunjukkan citra dengan 5 *blob* (b12, b30, b16, b27, b50) dan 3 *word* (gedung, langit, jalan), *blob* 50 dan 16 mewakili objek gedung, *blob* 30 mewakili objek jalan, dan *blob* 27 mewakili objek langit. *Blob* 12 tidak mewakili salah satu objek, seharusnya *blob* 12 mewakili *word* pohon –(tidak ada *word* pohon pada *keyword* citra gambar 3.3.b).

Beberapa hal yang terjadi dalam keterhubungan satu-satu antara *region* citra dengan notasi *keyword* pada data citra dalam database adalah sebagai berikut:

- 1. Jumlah *region* pada citra biasanya berbeda dengan jumlah *keyword*.

Gambar 3.3.a

2. Beberapa citra memiliki *region* yang tidak berhubungan dengan sejumlah *keyword*, ditunjukkan pada Gambar 3.3.b

Meskipun didapatkan notasi yang lengkap, pelatihan data untuk memperoleh keterhubungan antara *word* dan *region* masih juga masalah yang sulit. Karena *data set* tidak menyediakan keterhubungan yang eksplisit. Misal pada gambar 3.4.a menampilkan *tiger* di rumput, yang mempunyai *keyword* yang dinotasikan dengan *keyword tiger* dan *grass*, yang dapat diketahui adalah ada *tiger* dan rumput di citra, namun tidak dapat diketahui *region* yang mewakili objek *tiger* dan *region* yang mewakili objek rumput. Dengan citra tunggal, itu tidak dimungkinkan untuk memecahkan masalah ini. Jika ada *image* lain, dengan *region* berwarna *orange* (*region* yang berhubungan dengan *tiger*) dengan *region* yang lebih hijau (mewakili rumput), maka dimungkinkan untuk melakukan pelatihan data, *region* berwarna *orange* adalah *tiger*.

Masalah ini dapat dipecahkan dengan Algoritme EM, metode umum untuk memaksimalkan *likelihood* untuk mendasari distribusi data yang tidak lengkap atau hilang. Dalam kasus ini variabel yang hilang adalah keterhubungan antara *word* dan *region image*.

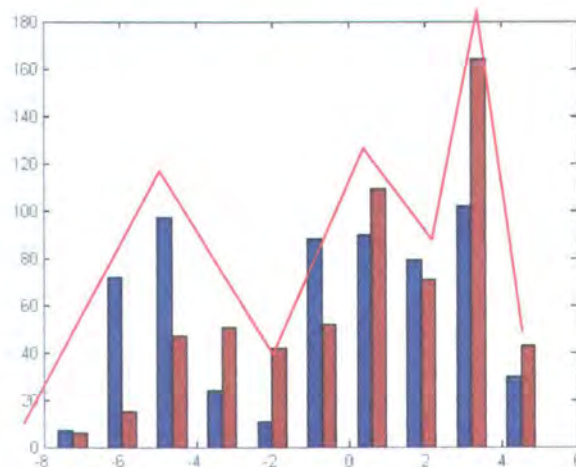
Pertama, *region* dengan terhubung dengan sekumpulan fitur dikumpulkan kedalam *single class* yang dinamakan *blob*. Kemudian membangun tabel kemungkinan yang menghubungkan antara *blob* dengan kata. Jika keterhubungan antara *blob* dan *word* dapat diperkirakan, maka itu kemungkinan untuk menghasilkan tabel probabilitas akan mudah.



Pada subbab 3.1.4.5 dijelaskan penggunaan algoritme *Expectation Maximization* dalam pelatihan data untuk mempelajari tabel probabilitas yang menghubungkan *blob* dengan kata. Dalam *test image*/pengenalan data citra baru yang diberikan, kata diprediksikan dalam setiap *blob* dengan menggunakan tabel probabilitas. Kemudian kata yang diprediksi akan digunakan dalam strategi prediksi *region naming* dan *auto annotation*.

#### 3.1.4.1 Dasar Teori Algoritme Expectation Maximization

*Mixture model* menggambarkan suatu bentuk representasi dari suatu data-data yang terkandung di dalam beberapa dokumen yang memiliki atribut atribut tertentu yang spesifik. Sebagai ilustrasi awal, misalkan terdapat distribusi dengan ruang sampel beranggotakan  $N$  yang diwakili oleh variable acak (*random variable*)  $X$ . Distribusi data  $N$  digambarkan pada gambar 3.5.



**Gambar 3.11 Contoh Distribusi *Mixture Model***

Dalam contoh distribusi tersebut ternyata ada tiga kelas besar, yang memiliki sifat-sifat tersendiri, dan tersembunyi. Namun data-data tersebut dapat diperkirakan masuk dalam kelompok tertentu dengan memperkirakan parameter mean dan varian distribusinya. Maka algoritme EM digunakan untuk melakukan estimasi terhadap parameter distribusi memodelkan dari data-data probabilistik sehingga data-data tersebut memiliki kemungkinan untuk masuk dalam suatu klaster tertentu.

Algoritme EM secara luas digunakan untuk memperkirakan parameter-parameter model menggunakan data yang tidak berlabel. Metode iteratif tersebut akan menghasilkan *Maximum Likelihood* (ML), yang menghasilkan parameter baru, yaitu bobot *mixture*, *means*, dan matrik *covarians*.

### 3.1.4.2 Maximum Likelihood

Diketahui fungsi densitas  $p(x | \Theta)$  yang diatur oleh parameter  $\Theta$  (misal  $p$  adalah *Gaussian* sedangkan  $\Theta$  terdiri dari *mean* dan *kovarians*), dan *data set* berukuran  $N$ , sehingga distribusi  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ . Data diasumsikan data vektor *independen*, dimana kejadian 1 *record* dan berikutnya tidak saling bergantung dan memiliki distribusi sama, yaitu distribusi  $p$ . Maka, *densitas/kerapatan* untuk sampel adalah:

$$p(\mathbf{X} | \Theta) = \prod_{i=1}^N p(x_i | \Theta) = L(\Theta | \mathbf{X}) \quad (3.10)$$

Fungsi  $L(\Theta | \mathbf{X})$  disebut fungsi *likelihood*, atau kemungkinan dari parameter-parameter data yang diketahui. *Likelihood* merupakan suatu fungsi dari parameter-parameter  $\Theta$  dimana data  $\mathbf{X}$  telah diketahui. Dalam persoalan *maximum*

*likelihood*, tujuan yang akan dilakukan adalah mencari  $\Theta^*$ , yaitu parameter  $\Theta$  yang memaksimumkan fungsi  $L$ , dimana:

$$\Theta^* = \arg \max_{\Theta} L(\Theta | X) \quad (3.11)$$

### 3.1.4.3 Dasar EM

EM merupakan metode umum untuk menemukan parameter-parameter *maximum likelihood* dari distribusi himpunan data yang diketahui, jika himpunan data tersebut *incomplete* atau mempunyai *missing value*. Data dikatakan *missing value* jika memiliki nilai yang hilang, *error*, atau ada nilainya tapi tidak berguna. Sedangkan suatu data disebut *incomplete* jika data tidak lengkap atau kacau karena keterbatasan proses observasi, misalnya tidak memasukkan parameter yang sebenarnya diperlukan dalam observasi. Pada kondisi kedua ini diasumsikan bahwa nilai parameter dianggap *hidden (missing)*.

Diasumsikan bahwa data  $X$  merupakan data observasi dan dihasilkan oleh beberapa distribusi. Data  $X$  disebut data yang tidak lengkap. Dimisalkan himpunan data  $Z$  yang lengkap adalah  $Z = (X, Y)$  dan fungsi densitas gabungannya yaitu:

$$p(z | \Theta) = p(x, y | \Theta) = p(y | x, \Theta)p(x | \Theta) \quad (3.12)$$

Fungsi densitas gabungan berasal dari fungsi densitas marginal  $p(x | \Theta)$  dan asumsi dari variabel tersembunyi atau nilai parameter yang diperkirakan. Pada keadaan ini  $Y$  merupakan informasi yang akan melengkapi data  $X$  sehingga menjadi data  $Z$  yang lengkap.

Dengan fungsi densitas baru maka dapat didefinisikan fungsi *likelihood* baru, yang disebut data *likelihood* yang lengkap.



$$L(\Theta | Z) = L(\Theta | X, Y) = p(X, Y | \Theta) \quad (3.13)$$

Fungsi ini adalah variabel random karena informasi yang hilang Y tidak diketahui, acak, dan diperkirakan dihasilkan oleh distribusi tertentu, sehingga:

$$L(\Theta | X, Y) = h_{X, \Theta}(Y) \text{ untuk beberapa fungsi } h_{X, \Theta}(\cdot)$$

dimana X dan  $\Theta$  adalah konstanta sedangkan Y merupakan variabel acak.

### 3.1.4.3.1 E-Step

Pada langkah *Expectation* dilakukan perhitungan nilai harapan (ekspektasi) dari *log-likelihood* data yang lengkap, yaitu  $\log p(X, Y | \Theta)$ .

Didefinisikan:

$$Q(\Theta, \Theta^{(i-1)}) = E[\log p(X, Y | \Theta | X, \Theta^{(i-1)})] \quad (3.14)$$

dimana  $\Theta$  = parameter baru yang akan dioptimasi untuk meningkatkan Q dan

$\Theta^{(i-1)}$  = parameter sekarang yang digunakan untuk evaluasi ekspektasi.

Berdasarkan informasi dari persamaan lain diperoleh:

$$E[(Y) | X = x] = \int_y h(y) f_{Y|X}(y | x) dy \quad (3.15)$$

dimana X dan  $\Theta^{(i-1)}$  adalah konstanta,  $\Theta$  merupakan variabel normal yang akan

disesuaikan, y adalah variabel acak yang diatur oleh distribusi  $f(y | X, \Theta^{(i-1)})$ .

Dengan menggunakan informasi tersebut ruas kanan dari persamaan (3.10) dapat ditulis sebagai berikut:

$$E[\log p(X, Y | \Theta | X, \Theta^{(i-1)})] = \int_{y \in Y} \log p(X, y | \Theta) f(y | X, \Theta^{(i-1)}) dy \quad (3.16)$$

dimana  $f(y | X, \Theta^{(i-1)})$  adalah distribusi marginal data bukan hasil pengamatan dan bergantung pada data pengamatan  $X$  dan parameter yang sekarang digunakan dan  $Y$  adalah ruang dari nilai-nilai  $y$ .

### 3.1.4.3.2 M-step

Langkah ini dilakukan dengan mencari argumen terakhir yang memaksimumkan fungsi tersebut dan dihentikan jika titik konvergensi tercapai (asumsi: telah melalui analisis konvergensi)

$$\Theta^{(i)} = \arg \max_{\Theta} Q(\Theta, \Theta^{(i-1)}) \quad (3.17)$$

dimana argumen  $\Theta$  merupakan parameter yang akan dioptimasi dalam usaha untuk memaksimumkan *likelihood* dan  $\Theta'$  merupakan parameter yang digunakan untuk evaluasi ekspektasi.

Kedua langkah E dan M diiterasi. Tiap iterasi dijamin untuk meningkatkan fungsi *log-likelihood*.

### 3.1.4.4 Parameter Maximum Likelihood pada densitas mixture

Masalah estimasi parameter *densitas mixture* merupakan salah satu aplikasi algoritme EM yang banyak digunakan dalam bidang pengenalan pola. Diasumsikan model probabilitas:

$$p(x | \Theta) = \sum_{i=1}^M \alpha_i p_i(x | \theta_i) \quad (3.18)$$

dimana parameter-parameter  $\Theta = (\alpha_1, \dots, \alpha_M, \theta_1, \dots, \theta_M)$ ,  $\sum_{i=1}^M \alpha_i = 1$ , tiap-tiap  $p_i$  adalah fungsi densitas dengan parameter  $\theta_i$ ,  $M$  komponen densitas dan *mixing coefficient*  $\alpha_i$ .

*Log-likelihood* data yang tak lengkap untuk densitas data  $X$ :

$$\log(L(\Theta | X)) = \log \prod_{i=1}^N p(x_i | \Theta) = \prod_{i=1}^N \log \left( \sum_{j=1}^M \alpha_j p_j \right) (x_i | \theta_j) \quad (3.19)$$

yang sulit dioptimasi karena terdapat penjumlahan *log*. Jika  $X$  dianggap *incomplete*, dimisalkan adanya data *item* yang tidak diamati  $Y = \{y_i\}_{i=1}^N$ , dimana nilainya menunjukkan densitas komponen yang menghasilkan tiap-tiap *item* data, dan persamaan *likelihood* dapat disederhanakan secara signifikan. Diasumsikan  $y_i \in 1, \dots, M$  untuk tiap  $i$  dan jika  $y_i = k$  jika *sample* ke- $i$  dihasilkan oleh komponen *mixture* ke- $k$ . Jika nilai  $Y$  diketahui, maka *likelihood* menjadi:

$$\begin{aligned} \log(L(\Theta | X, Y)) &= \log(p(X, Y | \Theta)) \\ &= \prod_{i=1}^N \log(p(x_i, y_i) p(y_i)) = \prod_{i=1}^N \log(\alpha_{y_i} p_{y_i}(x_i | \theta_{y_i})) \end{aligned} \quad (3.20)$$

Persoalannya, nilai-nilai dari  $Y$  tidak diketahui. Jika diasumsikan  $Y$  adalah vektor acak, maka dapat dilakukan proses berikutnya. Pertama, menurunkan persamaan untuk distribusi dari data yang tidak diamati. Diperkirakan parameter-parameter untuk densitas *mixture*, yaitu parameter yang tepat untuk *likelihood*  $L(\Theta^g | X, Y)$  dinyatakan dengan:

$$\Theta^g = (\alpha_1^g, \dots, \alpha_M^g, \theta_1^g, \dots, \theta_M^g) \quad (3.21)$$



Dengan diketahui  $\Theta^g$  dapat dihitung  $p_j(x_i | \theta_j^g)$  untuk tiap-tiap  $i$  dan  $j$ .

Sebagai tambahan, parameter *mixture*,  $\alpha_j$  dapat dianggap sebagai probabilitas *prior* dari tiap komponen *mixture*, sehingga  $\alpha_j = p(\text{komponen } j)$ . Selanjutnya dengan menggunakan aturan *Bayes* dapat diperoleh:

$$p(y_i | x_i, \Theta^g) = \frac{\alpha_{y_i} p_{y_i}(x_i | \theta_{y_i}^g)}{p(x_i | \Theta^g)} = \frac{\alpha_{y_i} p_{y_i}(x_i | \theta_{y_i}^g)}{\sum_{k=1}^M \alpha_k p_k(x | \theta_k^g)} \quad (3.22)$$

dan

$$p(x | \Theta) = \prod_{i=1}^N p(y_i | x_i, \Theta^g) \quad (3.23)$$

dimana  $y = (y_1, \dots, y_N)$  adalah contoh dari data yang tidak diamati. Berdasarkan persamaan-persamaan tersebut telah diperoleh densitas marginal yang diinginkan dengan mengasumsikan adanya variabel tersembunyi dan membuat perkiraan parameter-parameter awal dari distribusinya.

Dari persamaan (3.14), selanjutnya dapat dijabarkan dalam bentuk:

$$\begin{aligned} & Q(\Theta, \Theta^{(i-1)}) \\ &= \sum_{y \in Y} \log p(X, y | \Theta) p(y | X, \Theta^g) \\ &= \sum_{y \in Y} \log(L(\Theta | X, y)) p(y | X, \Theta^g) \\ &= \sum_{y \in Y} \sum_{i=1}^N \log(\alpha_{y_i} p_{y_i}(x_i | \theta_{y_i}^g)) \prod_{j=1}^N p(y_j | x_j, \Theta^g) \\ &= \sum_{y_1=1}^M \sum_{y_2=1}^M \dots \sum_{y_N=1}^M \sum_{i=1}^N \log(\alpha_{y_i} p_{y_i}(x_i | \theta_{y_i}^g)) \prod_{j=1}^N p(y_j | x_j, \Theta^g) \end{aligned}$$



$$= \sum_{y1=1}^M \sum_{y2=1}^M \dots \sum_{yN=1}^M \sum_{i=1}^M \sum_{l=1}^M \delta_{l,yi} \log(\alpha_l p_l(x_i | \theta_l)) \prod_{j=1}^N p(y_j | x_j, \Theta^g) \quad (3.24)$$

$$= \sum_{l=1}^M \sum_{i=1}^N \log(\alpha_l p_l(x_i | \theta_l)) \sum_{y1=1}^M \sum_{y2=1}^M \dots \sum_{yN=1}^M \delta_{l,yi} \prod_{j=1}^N p(y_j | x_j, \Theta^g) \quad (3.25)$$

Dari persamaan (3.24) dapat dijadikan ke bentuk persamaan (3.25) dengan contoh pembuktian di bawah ini.

Contoh:

$$\begin{aligned} & \sum_{y1=1}^2 \sum_{y2=1}^2 \sum_{i=1}^2 \sum_{l=1}^2 \delta_{l,yi} \log(\alpha_l p_l(x_i | \theta_l)) \prod_{j=1}^2 p(y_j | x_j, \Theta^g) \\ = & \{ \delta_{1,y1=1} \log(\alpha_1 p_1(x_1 | \theta_1)) + \delta_{1,y1=1} \log(\alpha_1 p_1(x_2 | \theta_1)) + \\ & \delta_{1,y1=2} \log(\alpha_1 p_1(x_1 | \theta_1)) + \delta_{1,y1=2} \log(\alpha_1 p_1(x_2 | \theta_1)) + \\ & \delta_{2,y1=1} \log(\alpha_2 p_2(x_1 | \theta_2)) + \delta_{2,y1=1} \log(\alpha_2 p_2(x_2 | \theta_2)) + \\ & \delta_{2,y1=2} \log(\alpha_2 p_2(x_1 | \theta_2)) + \delta_{2,y1=2} \log(\alpha_2 p_2(x_2 | \theta_2)) + \\ & \delta_{1,y2=1} \log(\alpha_1 p_1(x_1 | \theta_1)) + \delta_{1,y2=1} \log(\alpha_1 p_1(x_2 | \theta_1)) + \\ & \delta_{1,y2=2} \log(\alpha_1 p_1(x_1 | \theta_1)) + \delta_{1,y2=2} \log(\alpha_1 p_1(x_2 | \theta_1)) + \\ & \delta_{2,y2=1} \log(\alpha_2 p_2(x_1 | \theta_2)) + \delta_{2,y2=1} \log(\alpha_2 p_2(x_2 | \theta_2)) + \\ & \delta_{2,y2=2} \log(\alpha_2 p_2(x_1 | \theta_2)) + \delta_{2,y2=2} \log(\alpha_2 p_2(x_2 | \theta_2)) \} \\ & * \{ \prod_{j=1}^2 p(y_j | x_j, \Theta^g) \} \\ = & \{ \log(\alpha_1 p_1(x_1 | \theta_1)) (\delta_{1,y1=1} + \delta_{1,y1=2} + \delta_{1,y2=1} + \delta_{1,y2=2}) + \\ & \log(\alpha_1 p_1(x_2 | \theta_1)) (\delta_{1,y1=1} + \delta_{1,y1=2} + \delta_{1,y2=1} + \delta_{1,y2=2}) + \\ & \log(\alpha_2 p_2(x_1 | \theta_2)) (\delta_{2,y1=1} + \delta_{2,y1=2} + \delta_{2,y2=1} + \delta_{2,y2=2}) + \end{aligned}$$

$$\begin{aligned}
& \log(\alpha_2 p_2(x_2 | \theta_2))(\delta_{2,y1=1} + \delta_{2,y1=2} + \delta_{2,y2=1} + \delta_{2,y2=2})\} \\
& * \left\{ \prod_{j=1}^2 p(y_j | x_j, \Theta^g) \right\} \\
= & \{ \log(\alpha_1 p_1(x_1 | \theta_1))(1+0+1+0) + \log(\alpha_1 p_1(x_2 | \theta_1))(1+0+1+0) + \\
& \log(\alpha_2 p_2(x_1 | \theta_2))(0+1+0+1) + \log(\alpha_2 p_2(x_2 | \theta_2))(0+1+0+1) \} \\
& * \left\{ \prod_{j=1}^2 p(y_j | x_j, \Theta^g) \right\} \\
= & \{ 2 \log(\alpha_1 p_1(x_1 | \theta_1)) + 2 \log(\alpha_1 p_1(x_2 | \theta_1)) + \\
& 2 \log(\alpha_2 p_2(x_1 | \theta_2)) + 2 \log(\alpha_2 p_2(x_2 | \theta_2)) \} \left\{ \prod_{j=1}^2 p(y_j | x_j, \Theta^g) \right\} \\
= & \sum_{l=1}^2 \sum_{i=1}^2 \log(\alpha_l p_l(x_i | \theta_l)) \sum_{y1=1}^2 \sum_{y2=1}^2 \delta_{l,yi} \prod_{j=1}^2 p(y_j | x_j, \Theta^g)
\end{aligned}$$

Bentuk  $Q(\Theta, \Theta^{(i-1)})$  di atas terlihat cukup rumit, sehingga perlu disederhanakan, dengan catatan untuk  $l \in 1, \dots, M$ .

$$\begin{aligned}
& \sum_{y1=1}^M \sum_{y2=1}^M \dots \sum_{yN=1}^M \delta_{l,yi} \prod_{j=1}^N p(y_{yj} | x_j, \Theta^g) \\
= & \left( \sum_{y1=1}^M \dots \sum_{yi-1=1}^M \sum_{yi+1=1}^M \dots \sum_{yN=1}^M \delta_{l,yi} \prod_{j=1, j \neq i}^N p(y_{yj} | x_j, \Theta^g) \right) p(l | x_i, \Theta^g) \\
= & \prod_{j=1, j \neq i}^N \left( \sum_{y_j=1}^M p(y_j | x_j, \Theta^g) \right) p(l | x_i, \Theta^g) \\
= & p(l | x_i, \Theta^g)
\end{aligned} \tag{3.26}$$

karena  $\sum_{i=1}^M p(i | x_j, \Theta^g) = 1$ . Dengan menggunakan persamaan (3.26), maka

persamaan (3.25) dapat ditulis sebagai berikut:



$$\begin{aligned}
Q(\Theta, \Theta^{(i-1)}) &= \sum_{l=1}^M \sum_{i=1}^N \log(\alpha_l p_l(x_i | \theta_l)) p(l | x_i, \Theta^g) \\
&= \sum_{l=1}^M \sum_{i=1}^N \log(\alpha_l) p(l | x_i, \Theta^g) + \sum_{l=1}^M \sum_{i=1}^N \log(p_l(x_i | \theta_l)) p(l | x_i, \Theta^g) \quad (3.27)
\end{aligned}$$

#### 3.1.4.4.1 M-Step untuk Update Bobot

Untuk memaksimumkan persamaan (3.27), persamaan yang di dalamnya terdapat  $\alpha_l$  dan  $\theta_l$  secara independen dimaksimumkan, karena keduanya tidak berhubungan. Dalam menemukan ekspresi  $\alpha_l$ , dipergunakan *Lagrange multiplier*  $\lambda$  dengan batasan  $\sum_l \alpha_l = 1$ , dan menyelesaikan persamaan berikut:

$$\begin{aligned}
\frac{\delta}{\delta \alpha_l} \left[ \sum_{l=1}^M \sum_{i=1}^N \log(\alpha_l) p(l | x_i, \Theta^g) + \lambda \left( \sum_l \alpha_l - 1 \right) \right] &= 0 \\
\sum_{i=1}^N \frac{1}{\alpha_l} p(l | x_i, \Theta^g) + \lambda &= 0 \quad (3.28)
\end{aligned}$$

Dengan menjumlahkan kedua ukuran sampai melebihi  $l$ , diperoleh bahwa  $\lambda = -N$  yang menghasilkan persamaan untuk *update* nilai bobot *mixture* ( $\alpha$ ):

$$\begin{aligned}
\sum_{i=1}^N \frac{1}{\alpha_l} p(l | x_i, \Theta^g) - N &= 0 \\
\alpha_l &= \frac{1}{N} \sum_{i=1}^N p(l | x_i, \Theta^g) \quad (3.29)
\end{aligned}$$

#### 3.1.4.4.2 M-Step untuk Update Mean

Untuk beberapa distribusi dimungkinkan untuk memperoleh persamaan analitis untuk  $\theta$ , sebagai fungsi dari beberapa parameter. Sebagai contoh, untuk menurunkan persamaan distribusi *Gaussian Multivariate* (3.5) diperlukan beberapa sifat dari aljabar matriks.

Matriks kuadrat  $\text{tr}(A)$  sama dengan penjumlahan elemen-elemen diagonal  $A$ , yang memiliki sifat  $\text{tr}(A+B) = \text{tr}(A) + \text{tr}(B)$  dan  $\text{tr}(AB) = \text{tr}(BA)$ , yang menunjukkan bahwa  $\sum_i x_i^T A x_i = \text{tr}(AB)$  dimana  $B = \sum_i x_i x_i^T$ ,  $|A|$  merupakan determinan dari matriks  $A$  dan  $|A^{-1}| = 1/|A|$ . Diperlukan turunan dari fungsi matriks  $f(A)$  yang berhubungan dengan elemen-elemen matriks.

Selanjutnya didefinisikan  $\frac{\partial f(A)}{\partial A}$  untuk menjadi matriks dengan masukan ke- $i, j$   $\left[ \frac{\partial f(A)}{\partial a_{i,j}} \right]$  dimana  $a_{i,j}$  merupakan masukan ke- $i, j$  dari  $A$ . Definisi tersebut

juga diterapkan untuk turunan terhadap vektor. Pertama,  $\frac{\partial x^T A x}{\partial x} = (A + A^T)x$ .

Kedua, dapat ditunjukkan bahwa  $A$  disebut matriks simetrik jika:

$$\frac{\partial |A|}{\partial a_{i,j}} = \begin{cases} A_{i,j} & \rightarrow i = j \\ 2A_{i,j} & \rightarrow i \neq j \end{cases} \quad (3.30)$$

dimana  $A_{i,j}$  adalah kofaktor ke  $i, j$  dari  $A$ . Berdasarkan definisi di atas, dapat diperoleh:

$$\frac{\partial |A|}{\partial a_{i,j}} = \begin{cases} A_{i,j} / |A| & \rightarrow i = j \\ 2A_{i,j} / |A| & \rightarrow i \neq j \end{cases} = 2A^{-1} - \text{diag}(A^{-1}) \quad (3.31)$$

dengan definisi dari *inverse* suatu matriks. Akhirnya, dapat diperoleh :

$$\frac{\partial \text{tr}(AB)}{\partial A} = B + B^T - \text{diag}(B) \quad (3.32)$$

Berdasarkan *log* dari persamaan *Multivariate* (3.5), dengan konstanta diabaikan (karena akan hilang setelah diturunkan) dan disubstitusikan ke sisi kanan dari persamaan (3.27), sehingga diperoleh:

$$\begin{aligned} & \sum_{l=1}^M \sum_{i=1}^N \log(p_l(x_i | \theta_l)) p(l | x_i, \Theta^g) \\ &= \sum_{l=1}^M \sum_{i=1}^N \log \left( \frac{1}{(2\pi)^{d/2} |\Sigma_l|^{1/2}} e^{-\frac{1}{2}(x_i - \mu_l)^T \Sigma_l^{-1} (x_i - \mu_l)} \right) p(l | x_i, \Theta^g) \\ &= \sum_{l=1}^M \sum_{i=1}^N \left( -\frac{1}{2} \log(|\Sigma_l|) - \frac{1}{2} (x_i - \mu_l)^T \Sigma_l^{-1} (x_i - \mu_l) \right) p(l | x_i, \Theta^g) \end{aligned} \quad (3.33)$$

Turunan dari persamaan (3.32) yang berhubungan dengan  $\mu_l$ , diperoleh dengan turunan parsial terhadap  $\mu_l$  dan menyamakan persamaan sama dengan 0, sehingga diperoleh  $\mu_l$  sebagai berikut:

$$\begin{aligned} & \frac{\partial}{\partial \alpha_L} \left[ \sum_{l=1}^M \sum_{i=1}^N \left( -\frac{1}{2} \log(|\Sigma_l|) - \frac{1}{2} (x_i - \mu_l)^T \Sigma_l^{-1} (x_i - \mu_l) \right) p(l | x_i, \Theta^g) \right] = 0 \\ & \sum_{i=1}^N \Sigma_l^{-1} (x_i - \mu_l) p(l | x_i, \Theta^g) = 0 \\ & \sum_{i=1}^N x_i p(l | x_i, \Theta^g) = \sum_{i=1}^N \mu_l p(l | x_i, \Theta^g) \\ & \mu_l = \frac{\sum_{i=1}^N x_i p(l | x_i, \Theta^g)}{\sum_{i=1}^N p(l | x_i, \Theta^g)} \end{aligned} \quad (3.34)$$

### 3.1.4.4.3 Tahapan M untuk Update Kovarian



Untuk menemukan  $\Sigma_l$ , persamaan (3.34) dapat ditulis sebagai berikut:

$$\begin{aligned}
 & \sum_{l=1}^M \sum_{i=1}^N \left( -\frac{1}{2} \log(|\Sigma_l|) - \frac{1}{2} (x_i - \mu_l)^T \Sigma_l^{-1} (x_i - \mu_l) \right) p(l | x_i, \Theta^g) \\
 &= \sum_{l=1}^M \left[ \frac{1}{2} \log(|\Sigma_l^{-1}|) \sum_{i=1}^N p(l | x_i, \Theta^g) - \frac{1}{2} \sum_{i=1}^N p(l | x_i, \Theta^g) \text{tr}(\Sigma_l^{-1} (x_i - \mu_l)(x_i - \mu_l)^T) \right] \\
 &= \sum_{l=1}^M \left[ \frac{1}{2} \log(|\Sigma_l^{-1}|) \sum_{i=1}^N p(l | x_i, \Theta^g) - \frac{1}{2} \sum_{i=1}^N p(l | x_i, \Theta^g) \text{tr}(\Sigma_l^{-1} N_{l,i}) \right]
 \end{aligned}$$

dimana  $N_{l,i} = (x_i - \mu_l)(x_i - \mu_l)^T$

Dengan menghitung turunan parsial terhadap  $\Sigma_l^{-1}$  diperoleh:

$$\begin{aligned}
 & \frac{1}{2} \sum_{i=1}^N p(l | x_i, \Theta^g) (2\Sigma_l - \text{diag}(\Sigma_l)) - \frac{1}{2} \sum_{i=1}^N p(l | x_i, \Theta^g) (2N_{l,i} - \text{diag}(N_{l,i})) \\
 &= \frac{1}{2} \sum_{i=1}^N p(l | x_i, \Theta^g) (2M_{l,i} - \text{diag}(M_{l,i})) \\
 &= 2S - \text{diag}(S)
 \end{aligned}$$

dimana  $M_{l,i} = \Sigma_l - N_{l,i}$  dan  $S = \frac{1}{2} \sum_{i=1}^N p(l | x_i, \Theta^g) M_{l,i}$ .

Dengan menyamakan persamaan tersebut sama dengan nol,  $2S - \text{diag}(S) = 0$ ,

menunjukkan bahwa  $S = 0$ . Hal ini memberikan:

$$\sum_{i=1}^N p(l | x_i, \Theta^g) M_{l,i} = 0$$

$$\sum_{i=1}^N p(l | x_i, \Theta^g) (\Sigma_l - N_{l,i}) = 0$$

$$\sum_{i=1}^N p(l | x_i, \Theta^g) \Sigma_l = \sum_{i=1}^N p(l | x_i, \Theta^g) N_{l,i}$$

$$\Sigma_l = \frac{\sum_{i=1}^N p(l | x_i, \Theta^g) N_{l,i}}{\sum_{i=1}^N p(l | x_i, \Theta^g)}$$

$$\Sigma_l = \frac{\sum_{i=1}^N p(l | x_i, \Theta^g) (x_i - \mu_l)(x_i - \mu_l)^T}{\sum_{i=1}^N p(l | x_i, \Theta^g)} \quad (3.35)$$

### 3.1.4.5 Pelatihan Data Citra

#### 3.1.4.5.1 Pencarian keterhubungan antara *word* dan *blob*

Prediksi *word* pada *blob* dalam suatu citra diperoleh dari peluang tertinggi keterhubungan *word* dan *blob*.

$N$  : jumlah *item* (*image* dan *annotationnya*)

$M_T$  : jumlah kata dalam perbendaharaan kata

$L_T$  : jumlah semua *blob* dalam *data set*

$M_n$  : jumlah kata dalam  $n$ -buah *annotation*

$L_n$  : jumlah *blob* dalam  $n$ -buah *image*

$w_n$  : kata di  $n$ -buah *annotation*,  $w_n = (w_{n1}, \dots, w_{nj}, \dots, w_{nMn})$

$b_n$  : *Blob* di  $n$ -buah *image*,  $b_n = (b_{n1}, \dots, b_{nj}, \dots, b_{nMn})$

$w^*$  : Kata khusus

$b^*$  : *Blob* khusus

$t(w^* | b^*)$  : Kemungkinan dari dihasilkannya kata  $w^*$  yang diberikan pada *blob*  $b^*$

$a$  : sekumpulan keterhubungan *word* dan *blob*

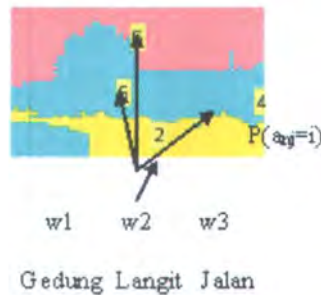
$a_n$  :  $a_{n1}, \dots, a_{nj}, \dots, a_{nMn}$  sehingga  $a_{nj} = i$  jika  $b_{ni}$  ditranslasikan ke  $w_{nj}$

$p(a_{nj} = i)$  : Kemungkinan penetapan  $a_{nj}$  adalah  $i$

$\theta$  : sekumpulan dari parameter data  $\theta = (p(a_{nj} = i), t(w_{nj} || b_{ni}))$

Tujuan penggunaan algoritma EM pada proses pencarian keterhubungan *word* dan *blob* ini adalah untuk mendapatkan hasil pemaksimalan  $p(w | b)$ , dengan  $b$  adalah sekumpulan *blob* dan  $w$  adalah sekumpulan *word*. Setiap kata dihubungkan dengan *blob* pada citra ditunjukkan pada Gambar 3.4. Model tersebut memerlukan seluruh kemungkinan/pejuang dari tiap pasangan dari *word* dan *blob* yang terhubung, sehingga  $p(w | b)$  dapat ditulis dalam notasi kemungkinan  $p(w, a|b)$  sebagai

$$p(w|b) = \sum_a p(w, a | b) \quad (3.36)$$



**Gambar 3.12 Peluang Keterhubungan *Word* dan *Blob***

Pada Gambar 3.5 citra tersebut memiliki 4 pasangan antara *blob* yaitu  $b_2$ ,  $b_5$ ,  $b_6$ ,  $b_4$  dan *word* dengan  $w_2$  mewakili objek langit, nantinya tiap-tiap *word* diprediksi nilai kemungkinan dengan tiap-tiap *blob*.

Jika citra mempunyai 1 *blob* dan  $m$  kata, maka peluang keterhubungan *word* dinyatakan dengan nilai yang spesifik  $a_j$  dengan  $j$  dari 1 sampai  $m$ . Jika *word* ke- $j$  dihubungkan dengan *blob* ke- $i$ , maka diperoleh notasi  $a_j=i$ . Sehingga  $p(w|b)$  dapat ditulis dengan

$$p(w | b) = \prod_{n=1}^N \prod_{j=1}^{Mn} \sum_{i=1}^{Ln} p(a_{nj} = i) t(w_{nj} | b_{(a_{nj}=i)}) \quad (3.37)$$



Sulit untuk memaksimalkan fungsi *likelihood* pada persamaan (3.37), karena ada fungsi penjumlahan dan perkalian di dalamnya. Masalah ini dapat diselesaikan dengan rumus *Expectation Maximization* dengan iterasi antara 2 tahap yang menghasilkan parameter  $\theta$  yang memaksimalkan *log-likelihood*  $Q^{ML}$ . Untuk masalah ini Fungsi  $Q^{ML}$  dapat dinyatakan dengan

$$Q^{ML} = \sum \sum \sum p(a_{nj} = i | w_{nj}, b_{ni}, \theta^{old}) \log [p(a_{nj} = i) t(w = w_{nj} | b = b_{(a_{nj}=i)})] \quad (3.38)$$

Fungsi dimaksimalkan pada batasan  $\sum_i p(a_{nj} = i) = 1$  untuk seluruh kata  $j$  dalam seluruh citra, dan  $\sum_{w^*} t(w^* | b^*) = 1$  untuk beberapa kata  $w^*$  dan setiap *blob*  $b^*$ . Selanjutnya persamaan  $t(w = w_{nj} | b = b_{(a_{nj}=i)})$  dapat ditulis dengan  $t(w_{nj} | b_{ni})$ .

Yang kemudian disempurnakan dengan pengenalan *Lagrangian*,  $L = Q^{ML} +$

$$\sum_{\alpha} \alpha \left( 1 - \sum_{i=1}^{L_n} p(a_{nj} = i) \right) + \sum_{b^*} \beta_{b^*} \left( 1 - \sum_{w^*} t(w^* | b^*) \right)$$

Dan dihitung turunannya dengan variabel  $(\alpha, \beta)$  dan parameter  $(p(a_{nj} = i) t(w^* | b^*))$ .

### 3.1.4.5.1 Tahap Expectation

1. untuk setiap  $n=1, \dots, N$  ;  $j=1, \dots, M_n$  dan  $i=1, \dots, L_n$  dihitung

$$\tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{old}) = p(a_{nj} = i) t(w_{nj} | b_{ni}) \quad (3.39)$$

2. Normalisasi  $\tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{old})$  untuk setiap citra ke-  $n$  dan kata ke-  $j$

$$\tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{old}) = \frac{\tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{old})}{\sum_{i=1}^{Ln} \tilde{p}(a_{nj} = i) t(w_{nj} | b_{ni})} \quad (3.40)$$

#### 3.1.4.5.2 Tahap Maximization

1. Untuk setiap pasangan yang berbeda  $(b^*, w^*)$  yang tampak bersama di setidaknya satu dari citra dihitung

$$\tilde{t}(w_{nj} = w^* | b_{ni} = b^*) = \sum_{n=1}^N \sum_{j=1}^{Mn} \sum_{i=1}^{Ln} \tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{old}) \delta_{(w^*, b^*)}(w_{nj}, b_{ni}) \quad (3.41)$$

Dengan  $\delta_{(w^*, b^*)}(w_{nj}, b_{ni})$  adalah 1, jika  $b^*$  dan  $w^*$  tampak pada citra dan 0 selain itu.

2. Normalisasi  $\tilde{t}(w_{nj} = w^* | b_{ni} = b^*)$  menghasilkan  $t(w_{nj} = w^* | b_{ni} = b^*)$

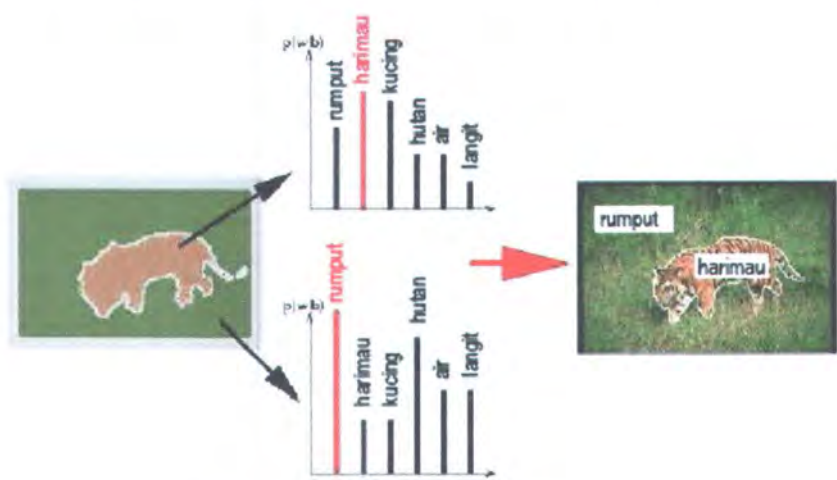
Algoritme EM digunakan untuk pelatihan yang mempelajari penggabungan tabel kemungkinan dari *word* dan *blob*. Diawali dengan membuat tabel probabilitas yang diberikan pada keterhubungan *blob* dan *word*. Kemudian tahap E dan M yang berulang sampai mencapai konvergen bertujuan untuk membangun perluang optimal dari tabel probabilitas yang menghubungkan *blob* dan *word*.

#### 3.1.4.5.3 Prediksi word

Tujuan dari prediksi *word*, adalah mencari keterhubungan *word* dengan setiap *blob* pada citra. Prediksi Data citra *test* dilakukan berdasarkan pd tabel probabilitas, memprediksi *word* untuk *region* citra (*region naming*). Ada 2

Strategi dalam memperkirakan keterhubungan *blob* dengan *word* pada suatu citra yaitu:

- 1. Strategi *Region Naming* adalah model untuk pengenalan objek, dengan memilih *word* terbaik untuk *region* menggunakan peluang  $p(w|b)$ , memilih peluang terbesar dari *word* untuk berkorespondensi pada *blob*.

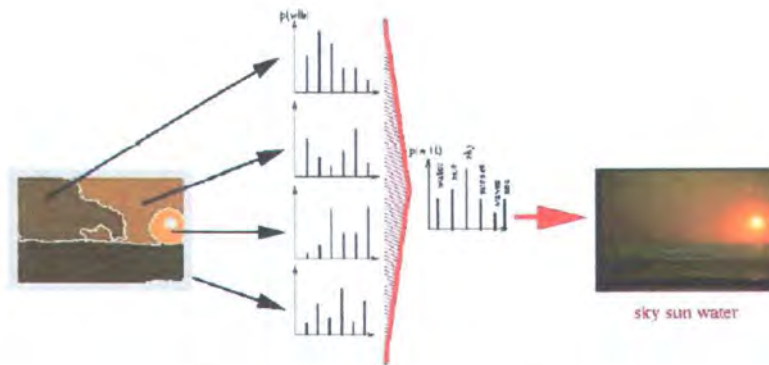


Gambar 3.13 *Region Naming*

Gambar 3.6 menunjukkan contoh yang menjelaskan strategi *region naming*. Untuk setiap *blob*, *word* dengan peluang terbesar dipilih dan dipilih dan diberikan pada *blob* tersebut.

- 2. Strategi *Auto-Annotation*, memprediksi *word* dari *blob* dengan menjumlahkan dan menormalkan keseluruhan peluang  $blob|word$  dalam citra tersebut. Notasi *n-word* didapatkan dari *n*-peluang terbesar table probabilitas, *n* adalah jumlah *keyword* aktual pada citra.





**Gambar 3.14 Auto-Annotation**

Gambar 3.7 menunjukkan peluang dari setiap *region* dijumlahkan dan dinormalisasi.  $n$  word dengan peluang terbesar dipilih sebagai *word* prediksi dari citra tersebut.

### 3.2 Pengenalan Data Citra Baru

Pada Tahap pengenalan Objek, Data citra baru tanpa notasi *word* diprediksi notasi *word* yang dimilikinya. Data citra baru disegmentasi menjadi *region-region* menggunakan algoritma *Mean Shift* dijelaskan pada bab 2, kemudian fitur dasar warna diekstrak dari masing-masing *region* dijelaskan pada subbab 3.1.2. Fitur yang diekstrak yaitu ukuran *region*, lokasi *region*, *mean* dan *varian* dalam domain warna *region*, luas *region* dibagi akar dari panjang *contour region*. Blob token untuk *region* data citra ini diketahui dengan menggunakan algoritma *nearest-neighbour*. Fitur dari *region* data citra baru dibandingkan dengan fitur dari *blob* pada data set menggunakan *Euclidian distance*. *Region* akan ditandai dengan *blob* yang paling dekat.

### 3.3 Pencarian Data Citra

Pengorganisasian data citra dijelaskan pada subbab 3.3.1, sedangkan strategi dalam memperoleh citra sesuai data input user dijelaskan pada subbab 3.3.2.

#### 3.3.1 Organisasi Data

Untuk memudahkan dalam pencarian, maka blob dan word perlu diorganisasikan. Dalam tahap pelatihan data dengan strategi auto-annotation diperoleh keterhubungan antara setiap citra dengan *word predict*. Telah dijelaskan pada sub bab 3.1.4.5.3, setiap citra dengan n-keyword mempunyai n-word predict. Setiap word predict mewakili kelas/objek pada citra. Gambar menunjukkan organisasi data dari objek langit, gedung dan burung. File citra yang memiliki objek langit juga beberapa juga memiliki objek gedung dan burung. File citra yang memiliki objek gedung juga memiliki objek langit. File citra yang memiliki objek burung juga memiliki objek langit. Desain data akan dijelaskan pada sub bab 4.1 dan 4.2.

Objek langit				
Citra				
Word predict	Gedung langit jalan	Gedung langit jalan	Gedung langit jalan	Gedung langit jalan
Citra				
Word predict	burung langit	burung langit	burung langit	burung langit
Objek Gedung				
Citra				
Word predict	Gedung langit jalan	Gedung langit jalan	Gedung langit jalan	Gedung langit jalan
Citra				
Word predict	Gedung langit	Gedung langit	Gedung langit	Gedung langit
Objek Burung				
Citra				
Word predict	Burung langit	Burung langit	Burung langit	Burung langit

Gambar 3.15 Kelas Objek Gedung , Langit, Burung pada Citra

3.3.2 Strategi Pencarian

User memasukkan data input berupa teks, setiap kata mewakili satu objek dari data citra. Data input teks ‘burung’ berarti menunjukkan pencarian dengan objek burung. Data input teks ‘langit’ berarti menunjukkan pencarian dengan



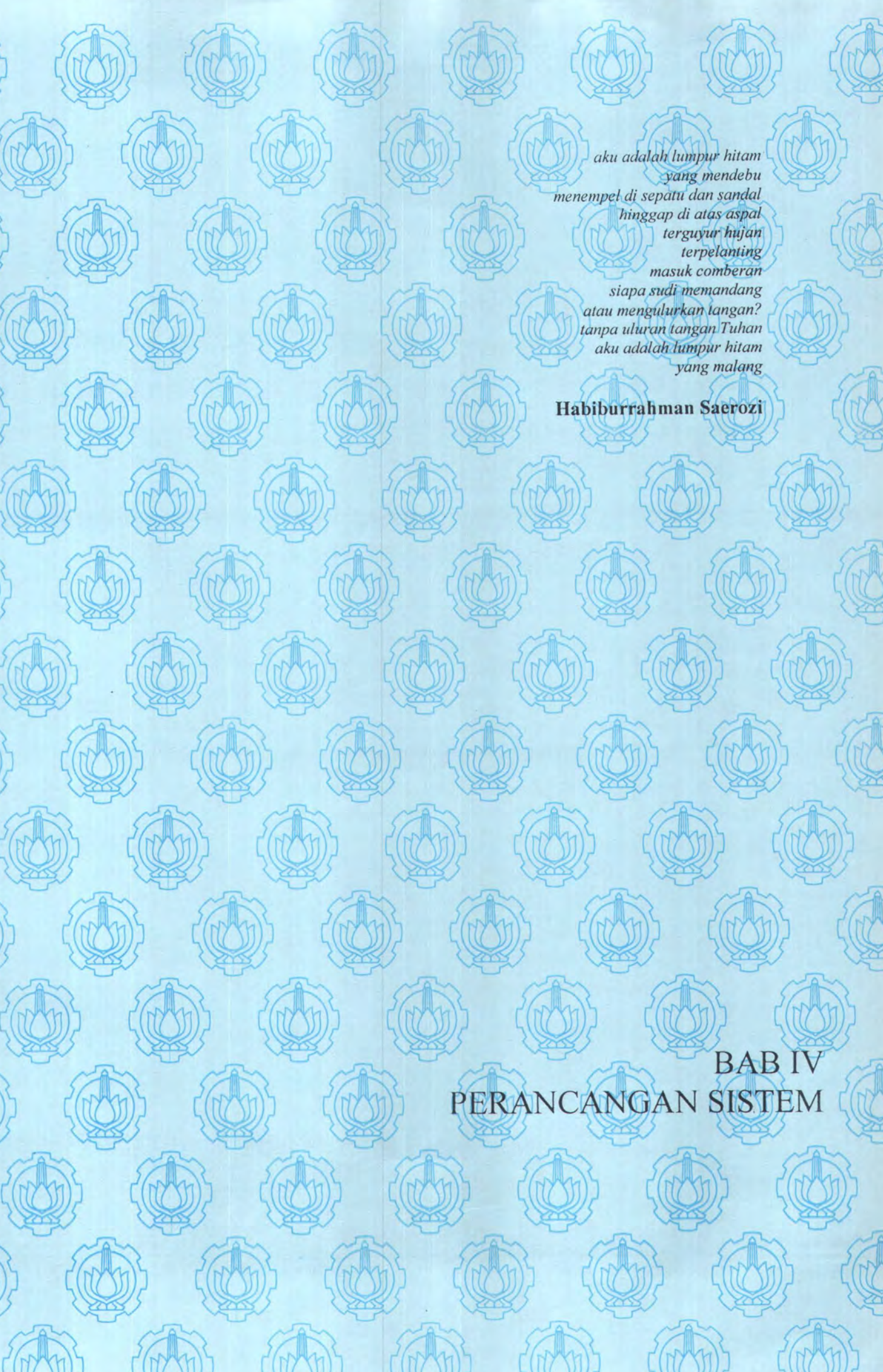
objek langit. Operator ‘AND’ dan ‘OR’ digunakan untuk menunjukkan pencarian lebih dari satu word objek.

Operator ‘OR’ menghasilkan citra-citra dengan satu atau dua objek yang tertulis pada data input, **Gambar** menunjukkan hasil dari operator langit dan . Gambar menunjukkan hasil data input ‘gedung’ AND ‘langit’. Operator ‘AND’ menunjukkan data input dengan 2 objek yang harus terdapat pada citra. Gambar menunjukkan hasil data input ‘gedung’ AND ‘langit’.

Objek Gedung				
Citra				
Word predict	Gedung langit jalan	Gedung langit jalan	Gedung langit jalan	Gedung langit jalan
Citra				
Word predict	Gedung langit jalan	Gedung langit jalan	Gedung langit jalan	Gedung langit jalan
Objek Burung				
Citra				
Word predict	Burung langit	Burung langit	Burung langit	Burung langit

Gambar 3.16 Pencarian pada Kelas Objek Gedung dan Burung





*aku adalah lumpur hitam  
yang mendebu  
menempel di sepatu dan sandal  
hinggap di atas aspal  
terguyur hujan  
terpelanting  
masuk comberan  
siapa sudi memandang  
atau mengulurkan langan?  
tanpa uluran tangan Tuhan  
aku adalah lumpur hitam  
yang malang*

**Habiburrahman Saerozi**

## **BAB IV PERANCANGAN SISTEM**



## **BAB 4**

### **PERANCANGAN SISTEM**

Dalam pembuatan perangkat lunak ini terdapat dua tahapan penting yaitu tahap perancangan desain dari perangkat lunak dan tahap implementasi perangkat lunak. Bab 4 ini menjelaskan aspek-aspek yang terdapat dalam kedua tahapan tersebut mulai dari tahap desain perangkat lunak yang meliputi desain data, desain proses, desain algoritme dan desain antar muka.

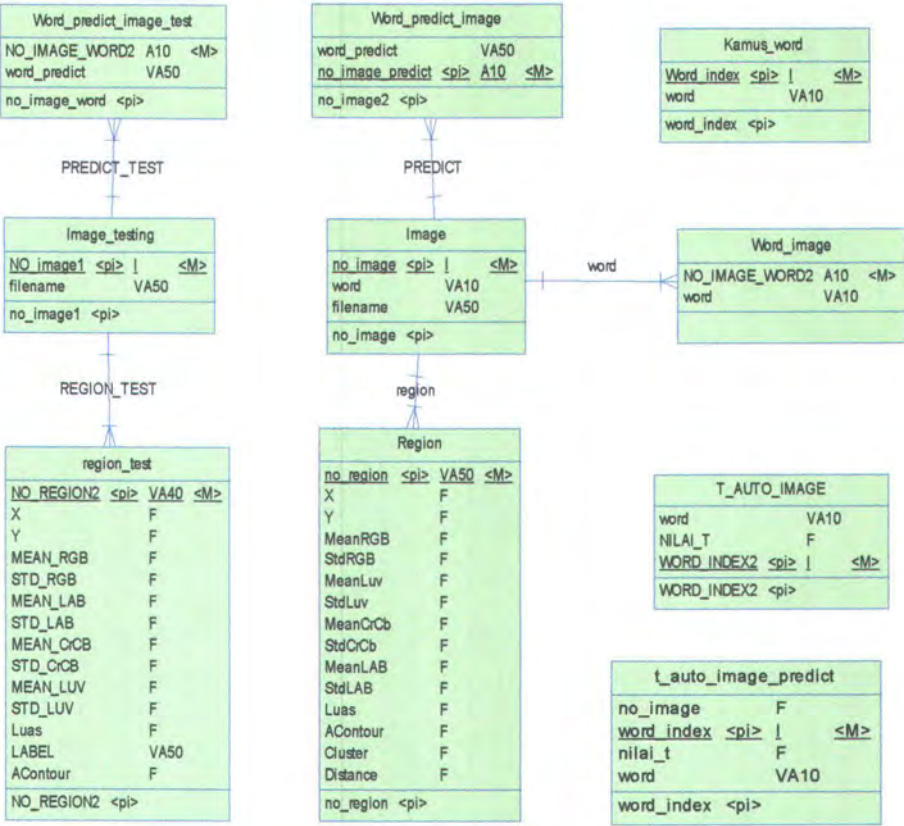
#### **4.1 Pemodelan Desain Data**

Desain data diperlukan untuk menentukan format data yang tepat bagi perangkat lunak hingga dapat dioperasikan secara benar. Terdapat tiga macam data yang diperlukan untuk pengoperasian perangkat lunak yaitu data masukan yang didapatkan dari pengguna perangkat lunak, data proses yang dibutuhkan dan dihasilkan selama proses eksekusi perangkat lunak dan data luaran yang memberikan hasil proses eksekusi perangkat lunak untuk pengguna yang menjalankannya.

##### **4.1.1 Desain Data CDM**

Conceptual Data Model adalah diagram yang menunjukkan hubungan antara entitas yang digunakan dalam desain data. Dalam desain data CBIR ini terdapat 10 entitas yaitu image/citra, image\_testing, word\_predict, word\_image, word\_predict\_image, region, region\_test, word\_predict\_image\_test, kamus\_word, t\_auto\_image, t\_auto\_image\_predict.





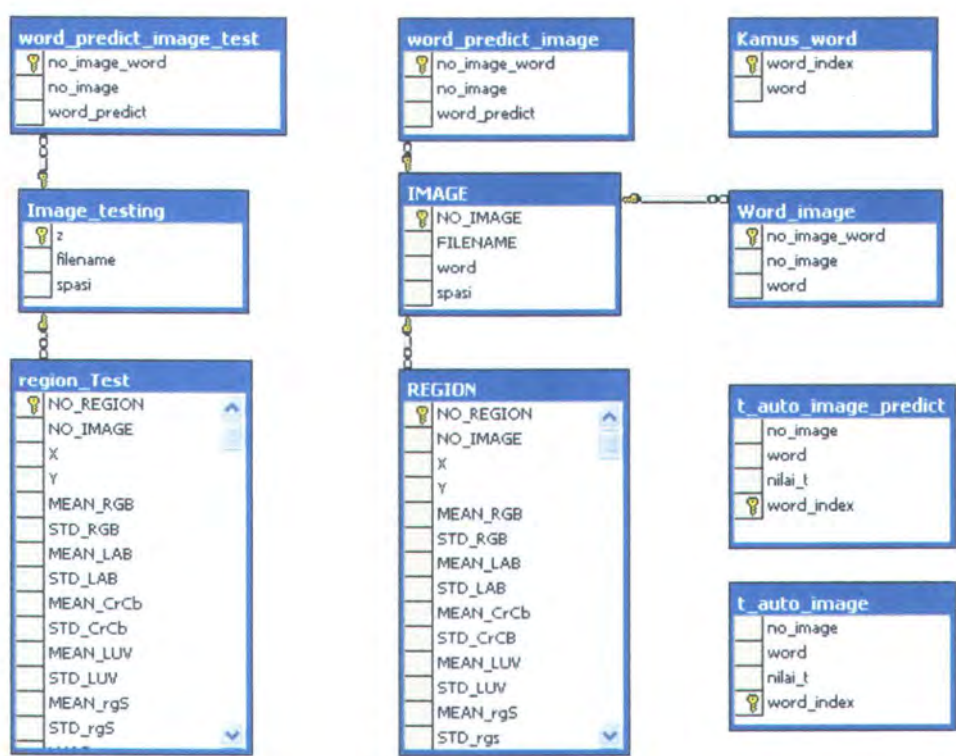
Gambar 4.17 Conceptual Data Model

Tabel 4.1 Keterangan CDM

No	Entitas	Primary key	Keterangan Entitas
1	image	no_image	Menyimpan data data citra pada pelatihan data citra
2	image_testin g	no_image1	Menyimpan data data citra pada pengenalan data citra baru
3	word_image	no_image_word	Menyimpan setiap word dari setiap image
4	word_predic t_image	no_image_predic t	Menyimpan word hasil prediksi dari pelatihan data citra
5	word_predic t_image_test	no_image_word2	Menyimpan word hasil prediksi dari pengenalan citra baru
6	Region	no_region	Menyimpan data vector fitur dari setiap region citra
7	Region_test	no_region2	Menyimpan data vector fitur dari setiap region citra baru
8	kamus_wor d	word_index	Menyimpan keseluruhan word
7	t_auto_imag e	word_index	Menyimpan nilai prediksi word dari citra
8	t_auto_imag e_predict	word_index2	Menyimpan nilai prediksi word dari citra baru

4.1.2 Desain Data PDM

Phisycal Data Model adalah diagram yang menunjukkan model data dan tabel dalam database, hasil dari penurunan Conseptual Data Model. Pada Model data ini penggunaan SQL/ Syntax Query Language dapat dilakukan. Pada PDM nama entitas pada CDM menjadi nama tabel dengan atribut yang sama dari CDM, satu primary key dan tambahan foreign key jika terdapat relasi one to many antara dua entitas.



Gambar 4.18 Physical Data Model

Tabel 4.2 Keterangan Phisycal Data Model

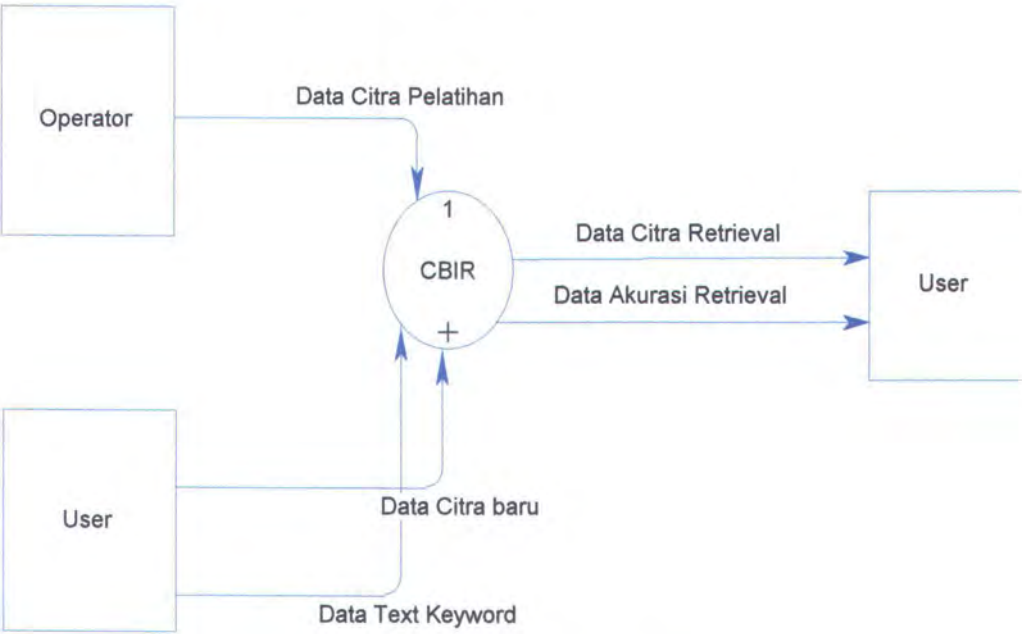
No	Field	Tabel	Type	Keterangan
1	No_image	Image, Word_image, TambahImage	Int	Nomer Citra
2	No_image_word	Word_image	Int	
3	Filename	Image	Varchar50	Path file citra disimpan
4	Word	Word_image, Kamus Word	Varchar50	Word dari citra
5	Word_index	Kamus Word	Int	Word dinotasikan dalam index
6	No_region	Region	Int	Nomer region [Image Region] [101] berarti no Image 10 region 1
7	X	Region	float	Centroid region X
8	Y	Region	float	Centroid region Y
9	MeanRGB	Region	float	Mean format RGB
10	StdRGB	Region	float	Varian format RGB
11	MeanLuv	Region	float	Mean format Luv
12	StdLuv	Region	float	Varian format Luv
13	MeanLab	Region	float	Mean format Lab
15	MeanCrCb	Region	float	Mean format CrCb
16	StdCrCb	Region	float	Varian format CrCb
17	Luas	Region	float	Luas region
18	Acontour	Region	float	Area / panjang contour
19	Cluster	Region	Int	Klaster region
20	Distance	Region	Float	Jarak antar fitur dengan data test



4.2 Pemodelan Proses

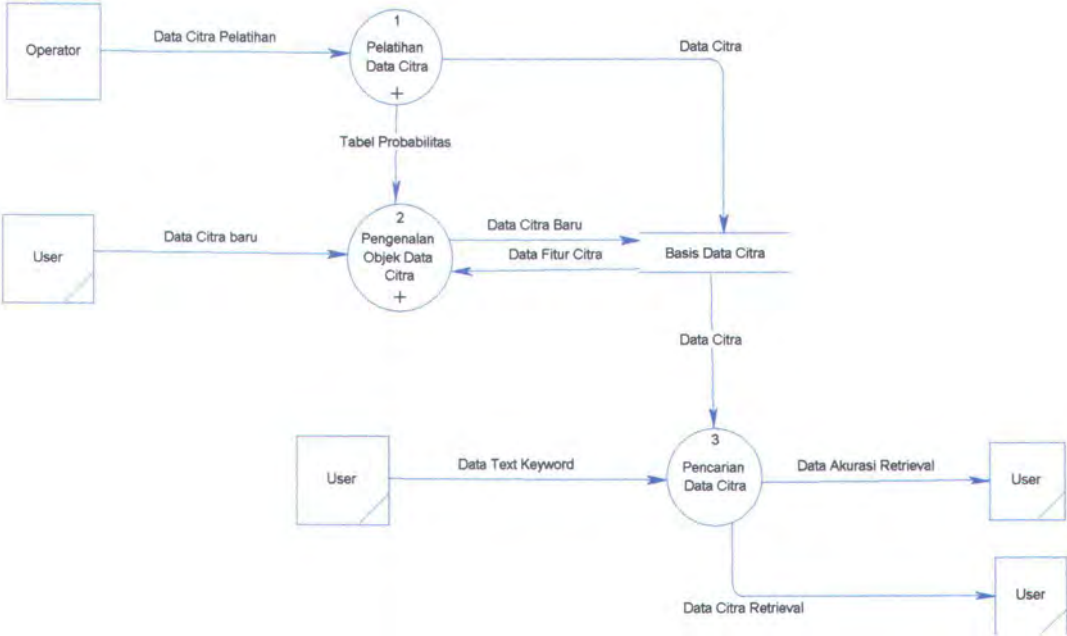
Diagram Aliran Data (*Data Flow Diagram*) adalah salah satu cara untuk menggambarkan proses yang dapat meperjelas perancangan dan pembuatan suatu perangkat lunak. Dari Diagram Aliran Data dapat diketahui data yang diolah dalam suatu proses dan data yang dihasilkan dari proses tersebut. Dengan demikian alur data dapat diketahui dengan jelas

4.2.1 DFD Level 0 Proses Utama dari CBIR



Gambar 4.19 DFD Level 0

4.2.2 DFD Level 1



Gambar 4.20 DFD Level 1

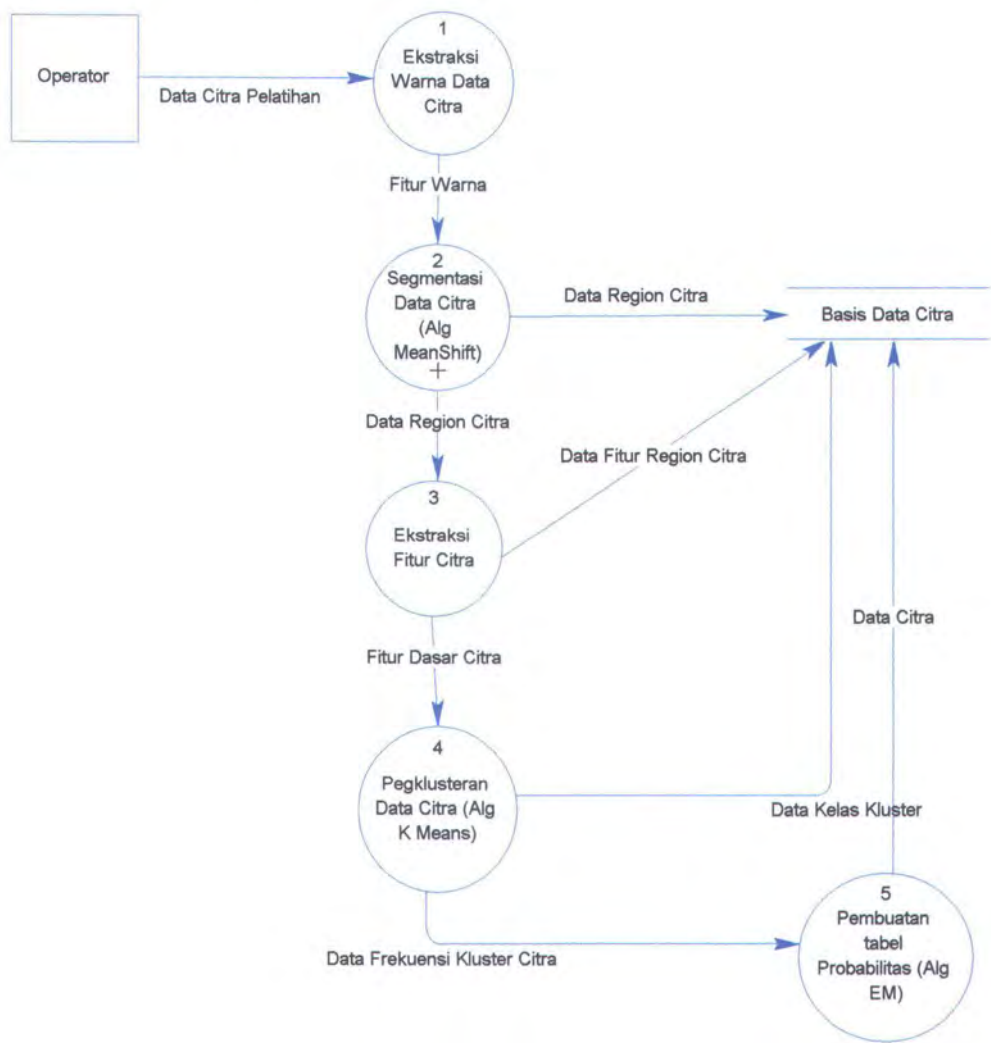
DFD level 1 menggambarkan keseluruhan proses secara lebih detail. Proses tersebut adalah sebagai berikut :

1. Pelatihan Data Citra, merupakan proses untuk melatih data citra yang terhubung dengan *text keyword* sehingga memiliki himpunan data *word* dan *blob*.
2. Pengenalan Objek Data Citra Baru, merupakan proses pengujian data, memasukkan data citra baru tanpa *text keyword*, dan memprediksi *word* dari *region* untuk setiap *blob* citra.
3. Pencarian Citra, melakukan pencarian citra berdasarkan teks yang dimasukkan oleh *user*.

4.2.2 DFD Level 2

Pada DFD level 2 dijelaskan tiga tahapan dalam CBIR yaitu pelatihan data citra, pengenalan data citra baru dan pencarian data citra.

4.2.2.1 DFD Level 2 Pelatihan Data Citra



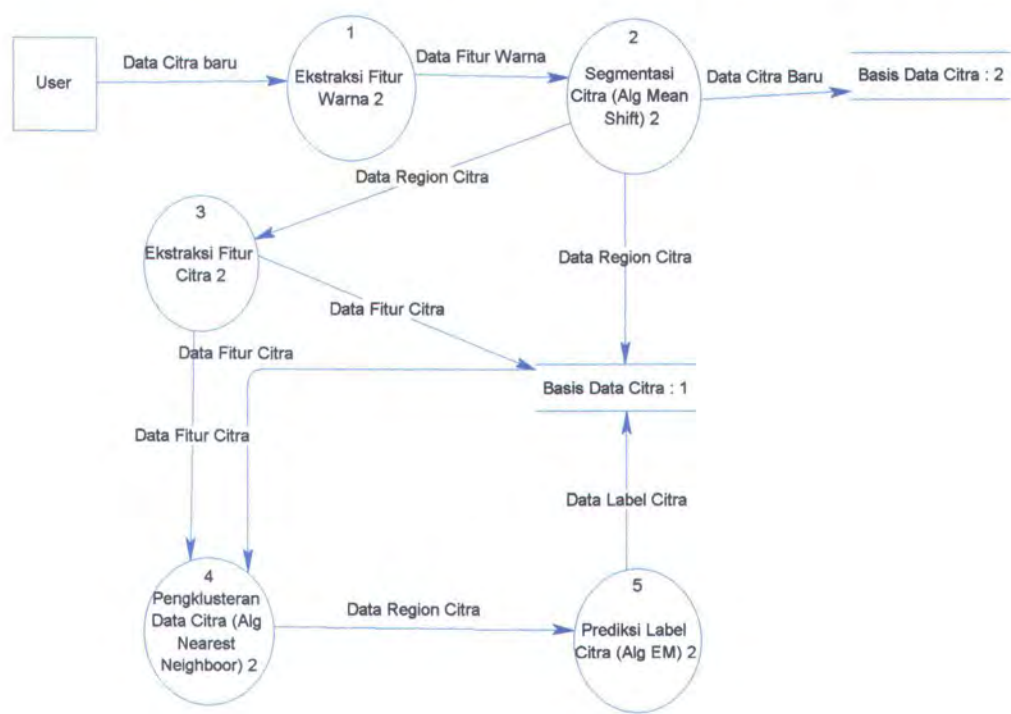
Gambar 4.21 DFD Level 2



**Sub proses pada proses pelatihan data adalah sebagai berikut:**

1. Ekstraksi Fitur Warna, Data citra dalam bentuk RGB yang dimasukkan ditransformasikan ke dalam bentuk fitur citra  $L*u*v$ .
2. Segmentasi *Mean Shift*, Citra pada data set disegmentasi menggunakan algoritme *Mean Shift* sehingga menjadi beberapa *region*.
3. Ekstraksi Fitur Dasar, mengekstraksi dari setiap *region* hasil segmentasi menjadi fitur-fitur ukuran *region*, lokasi *region*, luas *region* dibagi akar dari panjang *contour region* dan tekstur, mean dan varian dalam domain warna yang berbeda.
4. Pengelompokkan *region*, *region-region* dikelompokkan menjadi kelas-kelas yang disebut *blob* dengan algoritme *K-Means*. Setelah itu *blob-blob* yang sudah terbentuk ditandai (*tokenisasi*) menjadi *blob token*. Begitu juga dengan *keyword* sehingga menjadi *word token*. Setiap *region data set* diberi tanda, *region* tersebut termasuk *blob* yang mana, dengan memberikan label *blob token*. Begitu juga dengan *keyword*.
5. Pembuatan tabel Probabilitas dengan algoritme EM, menyusun tabel probabilitas mengenai kemungkinan bahwa suatu *blob* adalah suatu objek tertentu (atau *keyword* tertentu) digunakan algoritme *Expectation Maximization*, karena ada data yang '*missing*' yaitu korespondensi eksplisit yang menghubungkan antara *blob* dan *word*.

4.2.2.2 DFD Level 2 Pengenalan Objek Data Citra Baru



Gambar 4.22 DFD Level 2

Pada proses pengenalan objek data citra baru terdapat beberapa sub proses algoritme yang beberapa melalui sub proses yang sama pada proses pelatihan data:

1. Ekstraksi Fitur Warna, Data citra baru dalam bentuk RGB yang dimasukkan ditransformasikan ke dalam bentuk fitur citra  $L * u * v$ .
2. Segmentasi *Mean Shift*, Citra pada data set disegmentasi menggunakan algoritme *Mean Shift* sehingga menjadi beberapa *region*.

3. Ekstraksi Fitur Dasar, mengekstraksi dari setiap *region* hasil segmentasi menjadi fitur-fitur ukuran *region*, lokasi *region*, luas *region* dibagi akar dari panjang *contour region* dan tekstur, mean dan varian dalam domain warna yang berbeda.
4. Memasukkan data citra baru pada blob token yang memiliki ciri yang sama dengan menggunakan algoritme *Nearest-neighbor*. *Blob token* untuk *region* data tes ini diketahui dengan menggunakan algoritme *nearest-neighbor*. Fitur dari *region* data tes dibandingkan dengan fitur dari *blob* pada data set menggunakan *Euclidian distance*. *Region* akan ditandai dengan *blob* yang paling dekat.
5. Prediksi Label Citra dengan algoritme EM, memprediksi label/*keyword* pada citra baru berdasarkan kemungkinan/peluang tertinggi setiap *blob* dengan *word* pada data citra tersebut.

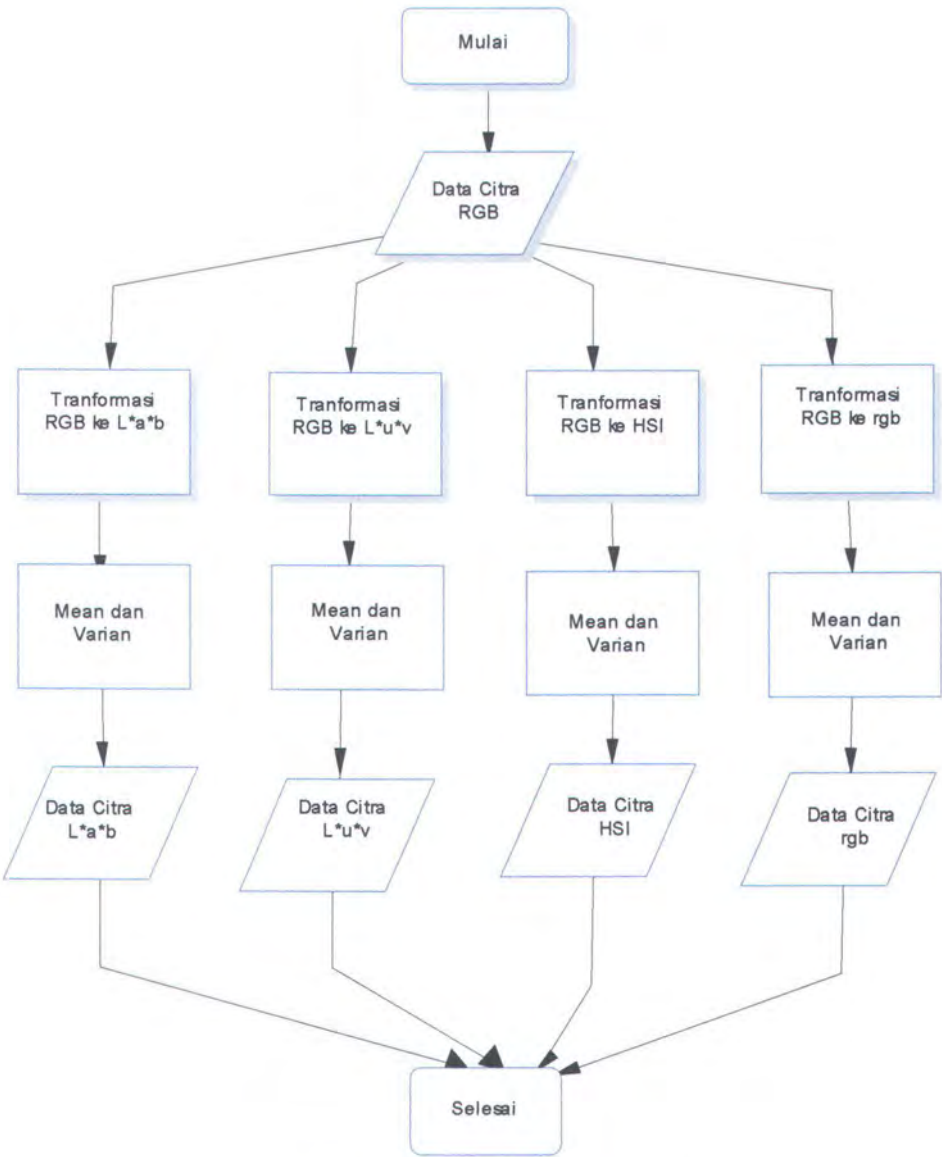
#### 4.3 Pemodelan Diagram Alur Algoritme

Sub Bab ini menjelaskan masing-masing tahapan algoritme : Ekstraksi Fitur Dasar Warna, Segmentasi *Mean Shift*, Klasterisasi *region K Means*, *Expectation Maximation*, *Nearest Neighbor*.



4.3.1 Algoritme Ekstraksi Fitur Dasar dan Warna

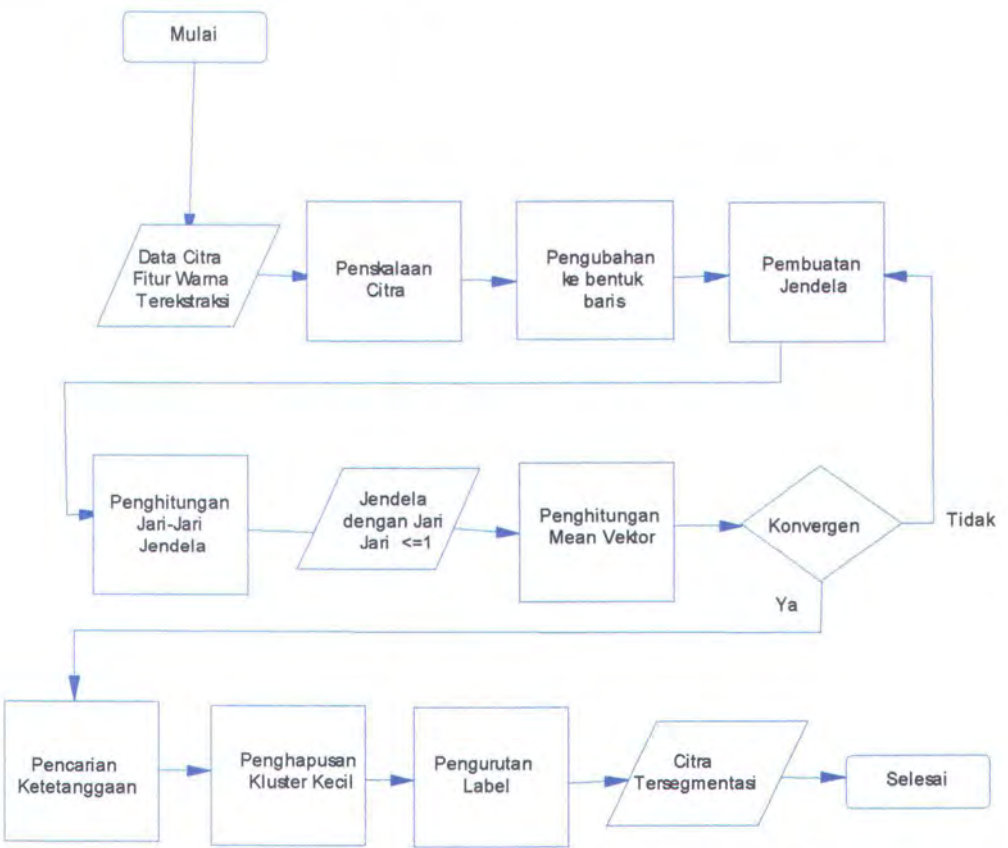
Algoritme yang menjelaskan tahapan ekstraksi fitur dasar berupa fitur warna dari sebuah citra RGB.



Gambar 4.23 Algoritme Ekstraksi Fitur

4.3.2 Algoritme *Mean Shift*

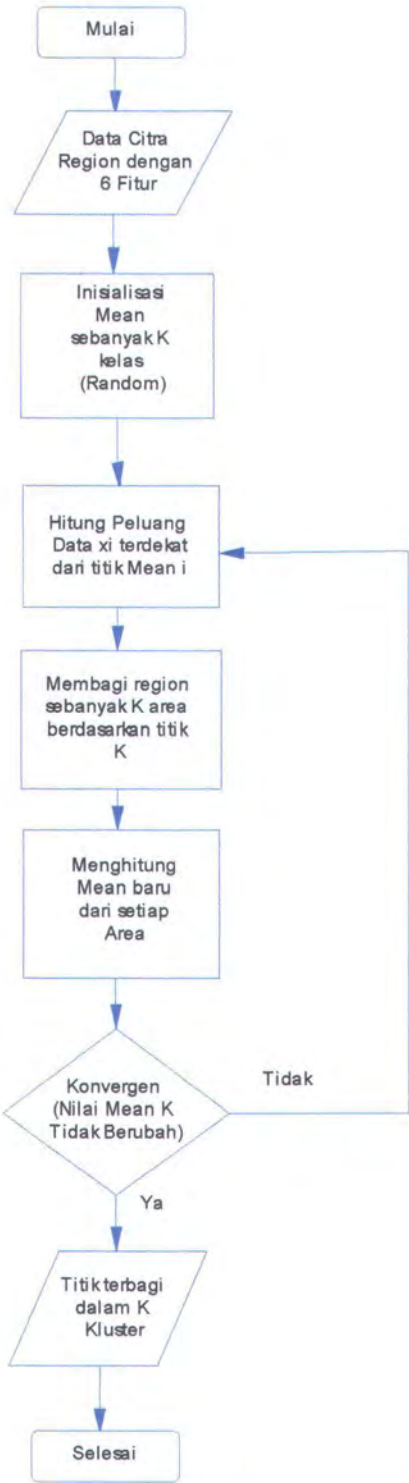
Algoritme *Mean Shift* bekerja pada bidang format warna *L* yang diambil dari format warna *L\*u\*v* dengan perpindahan jendela *Mean* sehingga mencapai titik konvergen (jarak perpindahan < 0.1)



Gambar 4.24 Algoritme Segmentasi *Mean Shift*

4.3.3 Algoritme *K-Means*

Algoritma Klasterisasi *K-Means* diawali dengan menentukan *K* titik *Mean* secara acak. Penentuan klas dari data-data berdasarkan pada jarak terdekat data-data tersebut dengan titik mean. Titik *Mean* akan berpindah hingga mencapai konvergen.



Gambar 4.25 Algoritme K-Means



#### 4.3.4 Algoritme *Nearest-Neighbor*

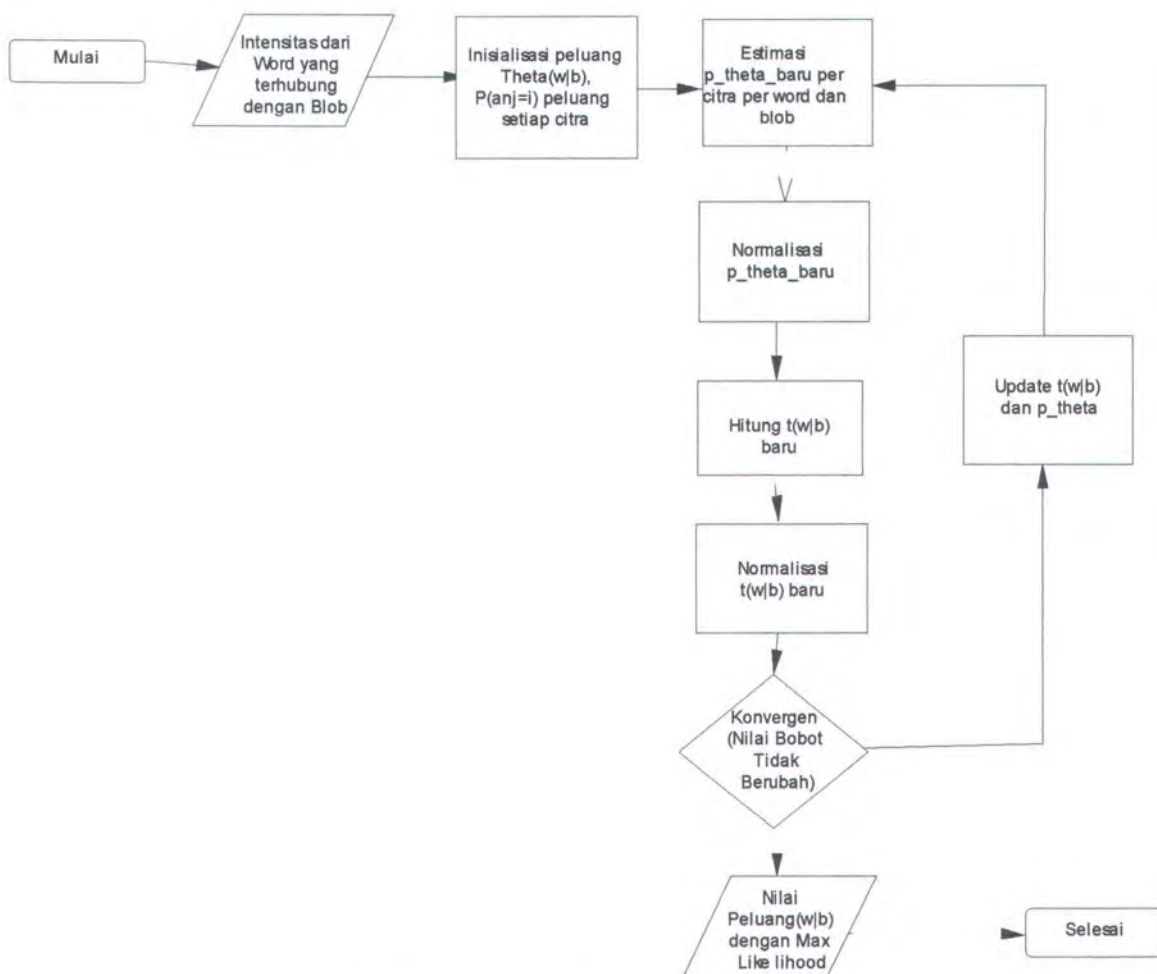
Algoritma untuk menentukan kelas *blob token* pada data citra baru dari *blob token* yang sudah ada sebelumnya, dengan berdasarkan pada kedekatan ciri-ciri / jarak *Euclidian* antara vector fitur data baru tersebut dengan *blob token* sebelumnya



Gambar 4.26 algoritme Nearest Neighbor

#### 4.3.5 Algoritme *Expectation Maximization*

Algoritma untuk menentukan peluang diberikannya *word* pada *blob* pada data citra, dengan memperkirakan ciri-ciri *Mean* dan kovarian dari distribusi data-data yang ada.

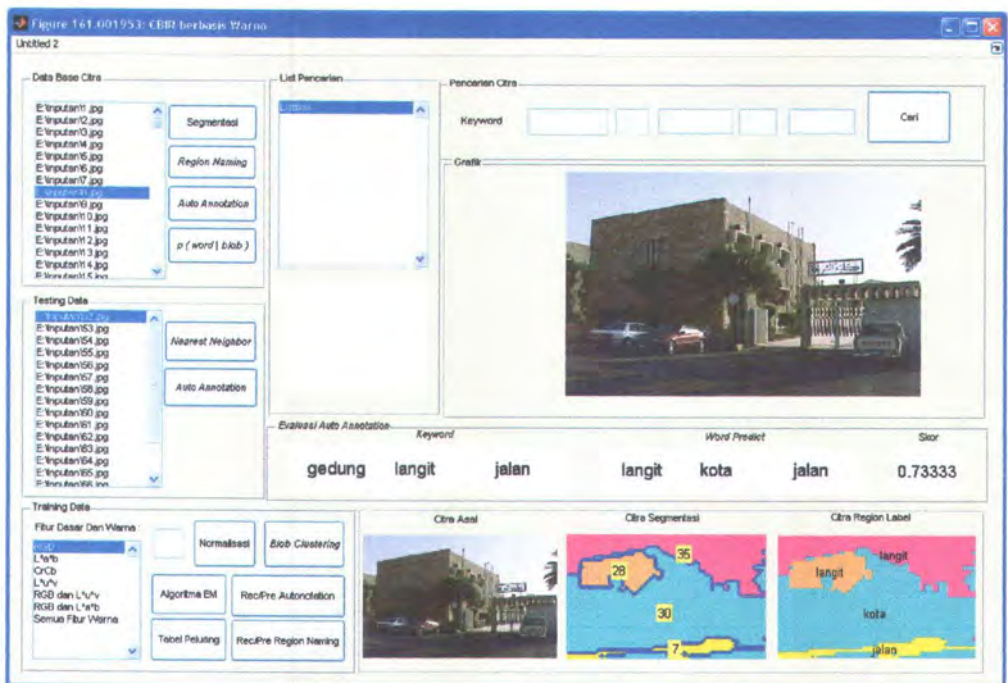


Gambar 4.27 Algoritma EM

#### 4.4 Pemodelan Antar Muka

Pemodelan Desain Antar muka perangkat lunak Content Based Image Retrieval ini mempunyai 3 *form* yaitu Database Citra, Pelatihan Data, dan Pencarian Citra. Fungsional dari masing-masing form adalah sebagai berikut:


1. Form Database Citra : List file citra, segmentasi, strategi *region naming*, strategi *auto-annotation*, dan menampilkan tabel peluang dalam satu citra
2. Form Pengenalan Data Citra Baru : List file citra baru, prediksi nearest neighbour, dan strategi *auto-annotation*.
3. Form Pencarian Data Citra : Teks Data Input, list hasil citra pencarian, dan jumlah data citra hasil pencarian



Gambar 4.28 Antarmuka perangkat lunak







*Bidadariku,  
Namamu tak terukir  
Dalam catatan harianku  
Asal usulmu tak hadir  
Dalam diskusi kehidupanku  
Wajah wujudmu tak terlukis  
Dalam sketsa mimpi-mimpiku  
Indah suaramu tak terekam  
Dalam pita batinku  
Namun kau hidup mengalir  
Pori-pori cinta dan semangatku  
Sebab  
Kau adalah hadiah agung  
Dari Tuhan*

**Habiburrahman Saerozi**

## **BAB V IMPLEMENTASI SISTEM**



## BAB 5

### IMPLEMENTASI SISTEM

#### 5.1 Implementasi *Mean Shift*

Sub proses segmentasi menggunakan algoritme *Mean Shift* dari data citra yang telah ditransformasikan dalam domain warna  $L^*u^*v$  memiliki beberapa tahapan sebagai berikut :

1. Penskalaan Citra, untuk mengatasi citra dengan ukuran besar yang akan membutuhkan waktu proses segmentasi yang lama maka data citra diskalakan dalam beberapa parameter skala yang sesuai. (1, ... 10 ) sesuai ukuran citra

```
1 Image = double(Image);
2 [NumRows,NumCols] = size(Image);
3 [x,y] = meshgrid(1:NumBaris,1:NumKolom);
4 TitikX = x(Spasi:Spasi:end-Spasi,Spasi:Spasi:end-Spasi);
5 TitikY = y(Spasi:Spasi:end-Spasi,Spasi:Spasi:end-Spasi);
```

2. Pengubahan ke bentuk baris, untuk mempermudah proses perkalian

```
1 TitikX = TitikX(:)';
2 TitikY = TitikY(:)';
3 TitikInds = sub2ind([NumBaris,NumKolom],TitikY,TitikX);
4 TitikI = Image(TitikInds(:))';
```

3. Pembuatan jendela, penelusuran mean vektor dilakukan dengan menggunakan jendela-jendela dengan ukuran  $m*m$  dari keseluruhan domain citra.

```
1 [NbrX,NbrY]=meshgrid(-CountSpasi-1:CountSpasi+1,-CountSpasi-1:CountSpasi+1);
2 NbrX = NbrX(:); NbrY = NbrY(:);
3 NumNbrs = length(NbrX); OneNbrs = ones(1,NumNbrs);
```

4. Perhitungan jari jari jendela, hanya jendela yang berjari-jari D kurang/ sama dengan 1 akan digunakan untuk perhitungan *Mean* Vektor.

```

1 Dists = [JendelaX - TitikX(1,OneNbrs(:) * (1:NumTitik));
2         JendelaY - TitikY(1,OneNbrs(:) * (1:NumTitik));
3         Intensitas - TitikI(1,OneNbrs(:) * (1:NumTitik))];
4 JarakSpasi = sqrt(sum(Dists(1:2,:).^2,1))/CountSpasi;
5 JarakIntensitas = abs(Dists(3,:))/CountInt;
6 JendelaPencari = JarakSpasi<1 & JarakIntensitas<1 & InView;
7 Normalization = sum(reshape(WithinThresh,NumNbrs,NumTitik),1);

```

5. Perhitungan *Mean* Vektor, menghitung nilai *mean* vektor dari jendela yang digunakan.

```

1 TitikX = sum(reshape(JendelaX .* WithinThresh, NumNbrs,NumTitik),1) ./ Normalization;
2 TitikY = sum(reshape(JendelaY .* WithinThresh, NumNbrs,NumTitik),1) ./ Normalization;
3 TitikI = sum(reshape(Intensitas .* WithinThresh, NumNbrs,NumTitik),1) ./ Normalization;

```

6. Berulang sampai konvergen

```

1 norm([TitikXLama,TitikYLama] - [TitikX,TitikY])>0 & Iter < MaxIterasi

```

7. Pencarian ketetanggaan, penghapusan kluster kecil, Menghapus nilai *region* dengan besar < Nmin. *Region* yang telah dilist dalam *List Region* akan tetap menjadi sebuah *region*, jika jumlah N vector fitur Luv > Nmin .Selain itu *region* dihapus dari list
8. Pengurutan label



## 5.2 Implementasi Ekstraksi Fitur

Sub Proses ini bertujuan mengesktraksi fitur dasar warna dari setiap region, nilai mean dan varian dari domain warna *domain*  $L^*u^*v$ ,  $L^*a^*b$ , RGB, CrCb dihitung dengan terlebih dahulu memetakan citra dari domain RGB ke dalam *domain*  $L^*u^*v$ ,  $L^*a^*b$ , RGB, CrCb .

### 5.2.1 Fitur Dasar

```
1 stats = regionprops(segmentasi,'all');
2 centroidX=stats(label).Centroid(1);
3 centroidY=stats(label).Centroid(2);
4 Area=stats(label).Area;
5 A_Contour= hitungA_per_contour(Area,segmentasi,label);
```

#### 5.2.1 Mean dan Varian

Fitur warna menggunakan rata-rata dan varian dalam domain warna RGB,  $L^*u^*v$ ,  $L^*a^*b$  dan CrCb

##### 5.2.1.1 nilai RGB

```
1 RGB=double(RGB);
```

##### 5.2.1.2 Domain CIE XYZ

```
1 X=0.431*RGB(:,:,1)+0.342* RGB(:,:,2) +0.178* RGB(:,:,3) ;
2 Y=0.222*RGB(:,:,1)+0.707* RGB(:,:,2) +0.071* RGB(:,:,3);
3 Z=0.020*RGB(:,:,1)+0.130*RGB(:,:,2)+0.939* RGB(:,:,3) ;
```

##### 5.2.1.3 Domain CrCb

```
1 RGB=double(RGB);
2 Y=16+(0.25878824*RGB(:,:,1)+0.50412941* RGB(:,:,2) +0.09790588* RGB(:,:,3)) ;
3 Cr=128+(-0.148229*RGB(:,:,1)-0.29099279* RGB(:,:,2) +0.43921589* RGB(:,:,3));
4 Cb=128+(0.4321589*RGB(:,:,1)-0.36778831*RGB(:,:,2)+-0.07142737* RGB(:,:,3)) ;
5 CrCb=cat(3,Cr,Cb);
```

#### 5.2.1.4 Domain L\*u\*v

```

1  RGB=double(RGB);
2  X=0.431*RGB(:,1)+0.342* RGB(:,2) +0.178* RGB(:,3) ;
3  Y=0.222*RGB(:,1)+0.707* RGB(:,2) +0.071* RGB(:,3);
4  Z=0.020*RGB(:,1)+0.130*RGB(:,2)+0.939* RGB(:,3) ;
5  white=[94.811 100.00 107.304];
6  [m n]=size(Y);
7  % Y = uint8(116*Y/white(2)^(1/3) - 16);
8  for i=1:m
9      For j=1:n
10         Y(i,j) = uint8(116*(Y(i,j)/white(2))^(1/3) - 16);
11     End
12 End
13 LUV=cat(3,cat(3,X,Y),Z);

```

#### 5.2.1.5 Domain CIE Lab

```

1  [m, n ,o ] = size(xyz);
2  for i=1:m
3      for j=1:n
4          if (xyz(i,j,2)/white(2) > 0.008856)
5              lab(i,j,1) = 116*(xyz(i,j,2)/white(2))^(1/3) - 16;
6          else
7              lab(i,j,1) = 903.3*(xyz(i,j,2)/white(2));
8          end
9      end
10 end
11 for k=1:m
12     for l=1:n
13         for i=1:3
14             if (xyz(k,l,i)/white(i) > 0.008856)
15                 fx(k,l,i) = (xyz(k,l,i)/white(i))^(1/3);
16             else
17                 fx(k,l,i) = 7.787*(xyz(k,l,i)/white(i)) + 16/116;
18             end
19         end
20     end

```

```

21 end
22 lab(:,2) = 500*abs(fx(:,1)-fx(:,2));
23 lab(:,3) = 200*abs(fx(:,2)-fx(:,3));

```

### 5.3 Implementasi Klasterisasi K Means

```

1 X=[X cell2mat(curs.Data(:,index_kolom:(index_kolom+1))));
2 X=[X cell2mat(curs.Data(:,15:16))];
3 % X=cell2mat(curs.Data(:,3:end-10));
4 no_region=curs.Data(:,1);
5
6 [cidx, ctrs] = kmeans2(X,8);%, 'dist','city', 'rep',100, 'disp','final');
7 marker={'g','o','r','x','c','+','m','*','y','s','k','d','v','^','<','>','p','h'};
8 for i=1:size(ctrs,1)
9     if(i>21)
10         plot(X(cidx==i,1), X(cidx==i,2), char(marker(i-19)));
11     end
12     if(i<=21)
13         hold on
14         plot(X(cidx==i,1), X(cidx==i,2), char(marker(i)));
15     end
16 end
17 plot(ctrs(:,1),ctrs(:,2),'rx');
18 hold off;

```

### 5.4 Implementasi *Expectation Maximization*

#### 5.4.1 Inisialisasi $t$ dan $p(a_{nj}=i)$

```

1 for n=1:N
2     [word]=get_word_image(n);
3     [blob]=get_region_image(n);
4     Ln=size(blob,2)-1;
5     Mn=size(word,1);
6     word_image(n,1)=n;
7     word_image(n,2)=Mn;
8     blob_image(n,1)=n;
9     for i=1:(size(blob,2))

```



```

10     blob_image(n,i+1)=blob(i);
11     end
12
13     for j=1:Mn
14         [word_i]=cari_w_i(word(j))
15         word_image(n,j+2)=word_i;
16         for i=1:Ln
17             p(n,i,j)=1/Ln;
18             t_intensitas(blob_image(n,i+2),word_i)=t_intensitas(blob_image(n,i+2),word_i)+1;
19         end
20     end
21 End

```

#### 5.4.2 Normalisasi t

```

1  for j=1:blob_total
2      for i=1:word_total
3          t(i,j)=t_intensitas(i,j)/sum(t_intensitas(i,:));
4      end
5  end

```

#### 5.4.3 Tahap E

1. Untuk setiap  $n=1, \dots, N$  ;  $j=1, \dots, M_n$  dan  $i=1, \dots, L_n$  dihitung

$$\tilde{p}(a_{nj} = i \mid w_{nj}, b_{ni}, \theta^{old}) = p(a_{nj} = i) t(w_{nj} \mid b_{ni})$$

```

1  for n=1:N
2      Mn=word_image(n,2);
3      Ln=blob_image(n,2);
4      for j=1:Mn
5          sum_p_theta0=0;
6          word_i=word_image(n,j+2);
7          for i=1:Ln
8              p_theta0(n,i,j)=p(n,i,j)*t(blob_image(n,i+2),word_i);
9              sum_p_theta0=sum_p_theta0+p_theta0(n,i,j);
10         end
11         sum_p_theta(n,j)=sum_p_theta0;
12     end
13 end

```

2. Normalisasi p\_theta untuk setiap image n dan kata j

$$\tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{old}) = \frac{p(a_{nj} = i | w_{nj}, b_{ni}, \theta^{old})}{\sum_{i=1}^{Ln} p(a_{nj} = i | w_{nj}, b_{ni})}$$

```

1  for n=1:N
2      Mn=word_image(n,2);
3      Ln=blob_image(n,2);
4      for j=1:Mn
5          for i=1:Ln
6              p_theta(n,i,j)=p_theta0(n,i,j)/sum_p_theta(n,j);
7          end
8      end
9  end

```

#### 5.4.4 Tahap M

1. Untuk setiap pasangan yang berbeda ( $b^*, w^*$ ) yang tampak bersama di setidaknya satu dari image, dihitung

$$\tilde{t}(w_{nj} = w^* | b_{ni} = b^*) = \sum_{n=1}^N \sum_{j=1}^{Mn} \sum_{i=1}^{Ln} \tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{old}) \delta_{(w^*, b^*)}(w_{nj}, b_{ni})$$

Dengan  $\delta_{(w^*, b^*)}(w_{nj}, b_{ni})$  adalah 1,

jika  $b^*$  dan  $w^*$  tampak di image / atau pernah terhubung  
dan 0 selain itu.

```

1  [a,b]=find(t_lama>0);
2  for wn=1:size(a,1)
3      sigma=0;
4      for n=1:N
5          Mn=word_image(n,2);
6          Ln=blob_image(n,2);
7          for j=1:Mn
8              for i=1:Ln
9                  if blob_image(n,i+2) == a(wn) && word_image(n,2+j)==b(wn)
10                     sigma=sigma+p_theta(n,i,j);
11                 tampil=[ n i j ]

```

```

12         end
13     end
14 end
15 end
16
17     t_lama(a(wn),b(wn))=sigma;
18 end

```

#### 5.4.5 Konvergen

Tahapan E dan M terus berulang sampai mencapai konvergen, jika Mencapai Nilai *Maximum Log Likelihood*.

$$Q^{ML} = \sum \sum \sum p(a_{\eta_j} = i \mid w_{\eta_j}, b_{\eta_i}, \theta^{old}) \log \left[ p(a_{\eta_j} = i) t(w = w_{\eta_j} \mid b = b_{(a_{\eta_j}=i)}) \right]$$

```

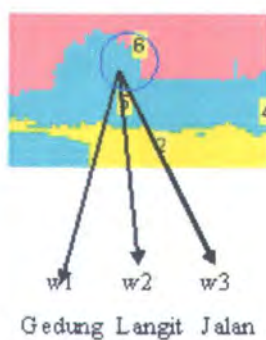
1 function
2 [max_LH]=hitung_max_like(word_image,blob_image,p_theta_baru,p_t_baru,N)
3 max_LH=0;
4 [a,b]=find(t_baru>0);
5 for wn=1:size(a,1)
6     sigmaN=0;
7     for n=1:N
8         Mn=word_image(n,2);
9         Ln=blob_image(n,2);
10        sigmaMn=0;
11        for j=1:Mn
12            sigmaLn=0;
13            for i=1:Ln
14                if blob_image(n,i+2) == a(wn) && word_image(n,2+j)==b(wn)
15                    sigmaLn=sigmaLn+p_theta_baru(n,i,j)*log(p(n,i,j)*t_baru(a(wn),b(wn)));
16                end
17            end
18            sigmaMn=sigmaMn+sigmaLn;
19        end
20    end
21    sigmaN=sigmaN+sigmaMn;
22 end

```



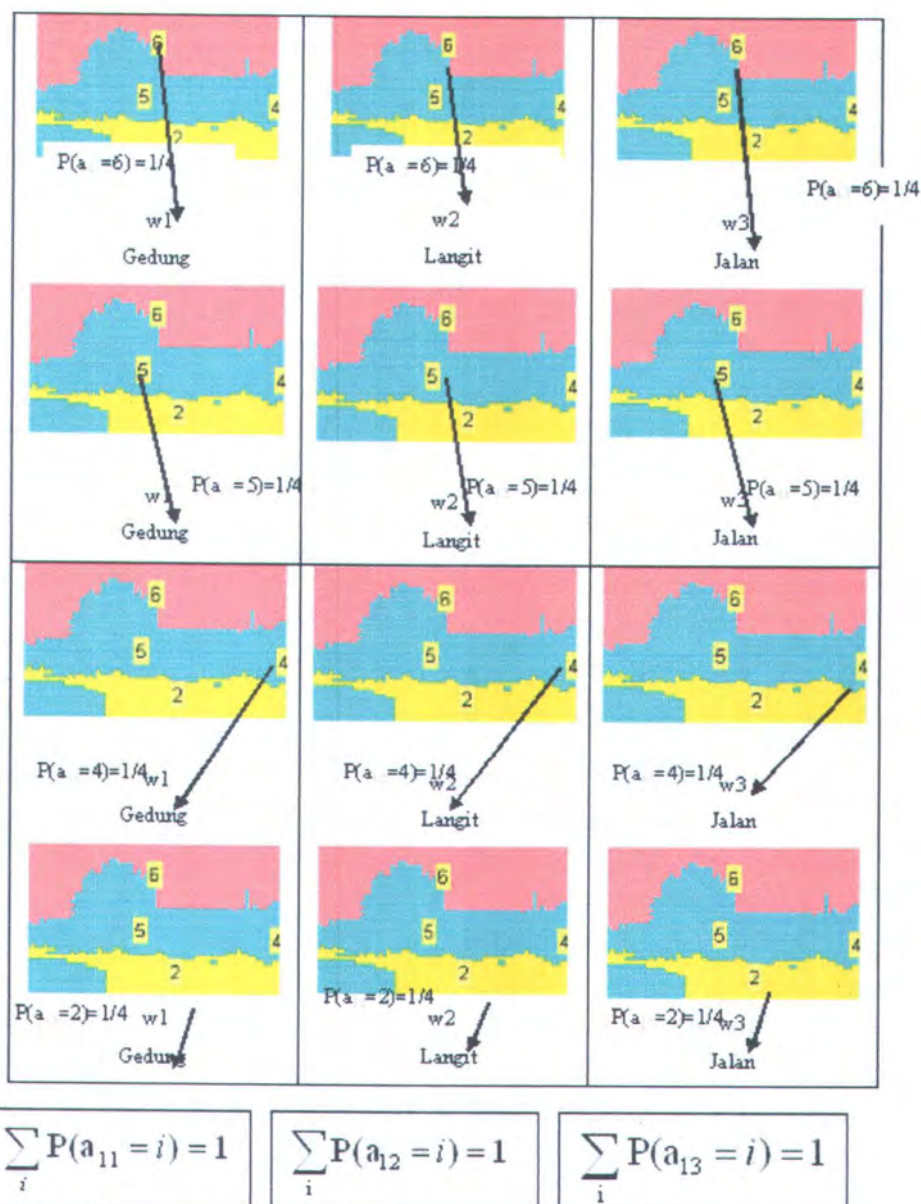
### 5.4.6 Penerapan EM

Pada subbab ini dijelaskan implentasi penerapan EM pada contoh citra no 1. Gambar 5.1 menunjukkan citra nomer 1 memiliki blob b yaitu b6, b5, b2, b4 dan himpunan word  $w$  = yaitu Gedung, jalan, langit. Notasi  $p(a_{nj}=i)$  adalah peluang keterhubungan antara *blob*  $i$  dengan *word*  $j$  pada citra- $n$ . Notasi  $t(w_{nj} | b_{ni})$  menyatakan frekuensi keterhubungan antar blob dan word.



**Gambar 5.29 Citra Penerapan EM**

### 5.4.6.1 Inisialisasi $p(a_{nj}=i)$



Gambar 5.30 Keterhubungan *Blob* dan *Word* dalam Suatu Citra

Inisialisasi  $p(a_{nj}=i)$  untuk setiap *image*  $N$ ,  $M_n$  dan  $L_n$





Tabel 5.5 peluang korespondensi *blob* pada citra 1 sampai 6 dengan *word* ke-3

P3	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	B <sub>5</sub>	B <sub>6</sub>	B <sub>7</sub>	B <sub>8</sub>	B <sub>9</sub> ..B <sub>13</sub>	$\sum_{w^*} t(w^*   b^*)$
n1	0.25	0.25	0.25	0.25	0	0	0	0	0	1
n2	0.25	0.25	0.25	0.25	0	0	0	0	0	1
n3	0.2	0.2	0.2	0.2	0.2	0	0	0	0	1
n4	0	0	0	0	0	0	0	0	0	1
n5	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0.125	0	1
n6	0	0	0	0	0	0	0	0	0	1
n7	0	0	0	0	0	0	0	0	0	1
...n23										

5.4.6.2 Inisialisasi t

$t(w_{nj} | b_{ni})$  adalah peluang keterhubungan *word*  $w_{nj}$  diberikan pada *blob*  $b_{ni}$

Inisialisasi Intensitas *Word/Blob* dalam seluruh citra.

Tabel 5.6 intensitas kehadiran *blob*-i dengan *word*-j

Intensitas Word/Blob	w1	W2	w3	w4	w5	w6	w7	w8	$\sum b   w$
b1	9	25	8	2	12	1	1	3	61
b2	3	23	2	1	17	3	1	3	53
b3	7	22	7	1	8	0	1	6	52
b4	10	20	9	2	7	1	0	2	51
b5	6	14	5	2	7	1	0	0	35
b6	0	15	0	2	11	1	1	2	32
b7	8	11	7	0	4	0	3	0	33
B8	7	10	7	0	3	1	0	0	28









Tabel 5.13 normalisasi p\_theta blob-i dengan word-1 pada citra 1 – 8

	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	B <sub>5</sub>	B <sub>6</sub>	B <sub>7</sub>	B <sub>8</sub>	B <sub>9</sub>	...B13
n1	0.1782	0.1941	0.2882	0.3396	0	0	0	0	0	
n2	0.1782	0.1941	0.2882	0.3396	0	0	0	0	0	
n3	0.1450	0.1489	0.1951	0.2346	0.2764	0	0	0	0	
n4	0	0	0	0	0	0	0	0	0	
n5	0.1144	0.0329	0.1175	0.1175	0.1540	0.1540	0.1247	0.1851	0	
n6	0	0	0	0	0	0	0	0	0	
n7	0.1386	0.1865	0.1865	0.2242	0.2642	0	0	0	0	
...n23										

5.4.6.4 Tahap M

Menghitung  $t(w_{nj} = w^* | b_{ni} = b^*)$

Tabel 5.14 keterhubungan blob dan word pada citra 1 sampai 8

No Image	Blob	Word	$\delta_{(w^*,b^*)}(w_{nj}, b_{ni})$
1	b6, b2, b5, b4	w1 , w2, w3	(b6-w1)=1,(b6-w2)=1, ..... (b4,w3)=1
2	b3, b6, b5, b2	w1 , w2, w3	(b3-w1)=1,(b3-w2)=1, ..... (b2,w3)=1
3	b6, b6, b5, b2, b2	w1 , w2, w3	(b6-w1)=1,(b6-w2)=1, ..... (b2,w3)=1
4	b4, b6, b6, b5, b2, b2, b1, b1	w1 , w2, w3	(b4-w1)=1,(b4-w2)=1, ..... (b1,w3)=1
5	b1, b1, b6, b5, b2	w1 , w2, w3	(b1-w1)=1,(b1-w2)=1, ..... (b2,w3)=1
6	b1, b1, b6, b6, b5, b2	w1 , w2, w3	(b1-w1)=1,(b1-w2)=1, ..... (b2,w3)=1
7	b1, b1, b3, b3, b3, b7, b2	W4, w2	(b1-w1)=1,(b1-w2)=1, ..... (b2,w3)=1
8	b5, b2, b4	W4, w2	(b1-w1)=1,(b1-w2)=1, ..... (b2,w3)=1

$\delta_{(w^*,b^*)}(w_{nj}, b_{ni})$  bernilai 0 pada **b6 → w4** dan **b7→ w1** ,

menyatakan pasangan blob dan word tersebut tidak pernah muncul dalam setidaknya pada satu citra. Untuk setiap pasangan  $w^*, b^*$  yang tampil setidaknya lebih pada 1 citra.

Tabel 5.15 normalisasi blob-i dengan word-1

	w1	w2	w3	w4	w5	w6	w7	w8
b1	1	1	1	1	1	1	1	1
b2	1	1	1	1	1	1	1	1
b3	1	1	1	1	1	0	1	1
b4	1	1	1	1	1	1	0	1
b5	1	1	1	1	1	1	0	0
b6	0	1	0	1	1	1	1	1
b7	1	1	1	0	1	0	1	0
b8	1	1	1	0	1	1	0	0

Tabel 5.16 nilai t pada citra 1 sampai 8

$b_i$	B <sub>1</sub>	B <sub>2</sub>	B <sub>3</sub>	B <sub>4</sub>	B <sub>5</sub>	B <sub>6</sub>	B <sub>7</sub>	B <sub>8</sub>	B <sub>9</sub>	...B13	$w_j$	w <sub>1</sub>	w <sub>2</sub>	w <sub>3</sub>
n1	1	5	7	8	0	0	0	0	0		n1	1	2	3
n2	1	5	7	8	0	0	0	0	0		n2	1	2	3
n3	1	3	4	7	8	0	0	0	0		n3	1	2	3
n4	1	2	4	5	6	0	0	0	0		n4	4	2	
n5	1	2	3	3	4	4	5	7	0		n5	1	2	3
n6	1	2	4	5	7	0	0	0	0		n6	1	0	
n7	1	3	4	5	6	0	0	0	0		n7	2	4	2
...n23											...n23			



Tabel 5.17 tabel t

	w1	w2	w3	W4	w5	w6	w7	w8
b1	1.3521	0	0	0	0	0	0	0
b2	0.1689	0	0	0	0	0	0	0
b3	0	0	0	0	0	0	0	0

Misal karena korespondensi yang ada adalah antara  
Nilai  $\delta_{(w^*,b^*)}(w_{nj},b_{ni})$  pada  $(w_{51}-b_{52} + w_{61}-b_{62} + w_{91}-b_{92})$  adalah 1, selain itu 0

$t(w_{n1} = w^* | b_{n2} = b^*) =$

$$\sum_{n=1}^N \sum_{j=1}^{Mn} \sum_{i=1}^{Ln} \tilde{p}(a_{nj} = i | w_{nj}, b_{ni}, \theta^{old}) \delta_{(w^*,b^*)}(w_{nj}, b_{ni})$$

$= \tilde{p}(a_{51} = 2 | w_{51}, b_{52}, \theta^{old}) + \tilde{p}(a_{61} = 2 | w_{61}, b_{62}, \theta^{old}) + \tilde{p}(a_{91} = 2 | w_{91}, b_{92}, \theta^{old})$

$= 0.0442 + 0.0695 + 0.0551 +$

$= 0.1689$

Tabel 5.18 normalisasi t blob-i dengan word-j pada citra 1 – 8

t(Word/Blob)	W1	w2	w3	w4	w5	w6	w7	w8	$\sum_{w^*} t(w^*   b^*)$
b1	0.1520	0.4684	0.1280	0.0350	0.1768	0.0061	0.0051	0.0286	1
b2	0.0225	0.4317	0.0097	0.0119	0.4031	0.0751	0.0070	0.0390	1
b3	0.1292	0.4502	0.1415	0.0140	0.0860	0	0.0082	0.1709	1
b4	0.2294	0.4116	0.1950	0.0474	0.0916	0.0083	0	0.0166	1
b5	0.1840	0.4129	0.1351	0.0940	0.1577	0.0164	0	0	1
b6	0	0.4291	0	0.0924	0.4164	0.0161	0.0136	0.0324	1
b7	0.3140	0.2583	0.2573	0	0.0533	0	0.1170	0	1
b8	0.3134	0.2904	0.3406	0	0.0342	0.0215	0	0	1

## 5.5 Implementasi *Nearest-Neighbor*

Sub proses Algoritme *Nearest Neighbor* memiliki beberapa tahapan :

1. Perhitungan jarak antara titik baru dengan titik yang sudah ada, menghitung jarak *euclidian* antara tiap N data titik lama dengan titik baru  
Data Citra lama dan baru memiliki 6 fitur dasar (  $x_1, \dots, x_6$  ), sehingga  
Jarak *euclidian* =

$$\sqrt{(new\_x_1 - x_1)^2 + (new\_x_2 - x_2)^2 + \dots + (new\_x_6 - x_6)^2}$$

```

1  for i=1:row
2      Sum_jarak=0;
3      for j=1:col-1
4          Sum_jarak=Sum_jarak+((Objek1(i,j)-Objek2(j))^2);
5      End
6      jarak(i)=sqrt(Sum_jarak);
7  End

```

2. Mengurutkan jarak, *sort(jarak\_euclidian)*, Pengambilan k-elemen teratas, jika nilai k=8 , maka diambil 8 jarak terdekat dari titik baru terhadap titik-titik lama, menghitung jumlah jelas data dari k-elemen, dan mengambil jumlah kelas terbanyak menjadi kelas data bagi titik baru

```

1  for i=1:row
2      colnames={'DISTANCE'};
3      exdata = {jarak(i)};
4      whereclause='where NO_REGION = ';
5      whereclause=strcat(whereclause,char(no_region(i)));
6      update(conn, 'region', colnames, exdata, whereclause );
7  End
8  curs = exec(conn, 'SELECT TOP 1 CLUSTER,COUNT(CLUSTER) FROM (select TOP 8
no_region,distance,cluster from region order by distance asc) a GROUP BY CLUSTER')

```



Thaaha.  
Maa anzalna 'alaikal Qur'aana li tasyqa  
Illa tadzkiratan liman yakhsya  
(Kami tidak menurunkan Al-Qur'an ini  
kepadamu  
agar kamu jadi susah  
Tetapi sebagai tadzkirah  
bagi orang yang takut kepada Allah)

QS. Thaaha: 1-3.

## BAB VI UJI COBA DAN EVALUASI



## **BAB 6**

### **UJI COBA DAN EVALUASI**

Dalam bab ini dibahas mengenai uji coba CBIR menggunakan dua tahap pengujian perangkat lunak berbasis objek beserta evaluasi hasil. Subbab 6.1 menjelaskan tentang data input yang digunakan dalam uji coba. Subbab 6.2 menjelaskan tentang metode pengukuran kinerja CBIR. Subbab 6.3 menjelaskan tentang evaluasi uji coba kinerja CBIR dengan parameter jumlah iterasi. Subbab 6.4 menjelaskan tentang evaluasi uji coba kinerja CBIR dengan parameter jumlah kelas/blob. Subbab 6.5 menjelaskan tentang evaluasi uji coba kinerja CBIR dalam penggunaan fitur dari domain warna yang berbeda. Subbab 6.5 menjelaskan tentang evaluasi uji coba pada tahap pengenalan data citra baru, sedangkan 6.6 uji coba pada tahap pencarian data citra .

#### **6.1 Data Input**

##### **6.1.1 Penggunaan Corel Data Set sebagai data input**

Corel data set digunakan sebagai data input, Corel data set dapat diperoleh dalam cd. Beberapa nama dan topik dalam cd tersebut adalah :

1. Gedung Arsitektur
2. Tempat – tempat penting di dunia
3. Kota dan negara
4. Binatang
5. Pemandangan

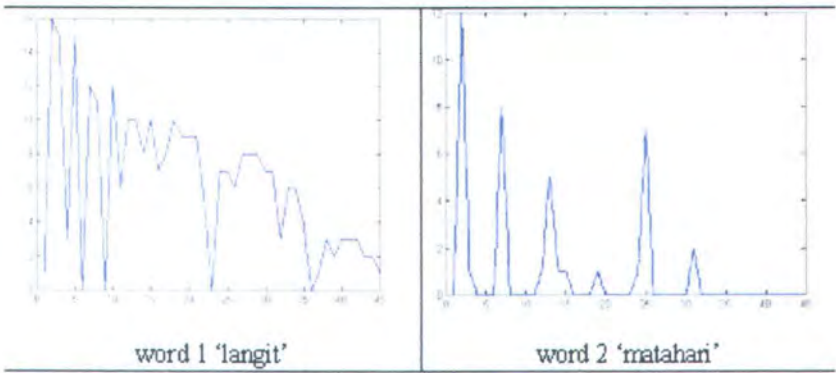
- 6. Laut
- 7. Peralatan



Gambar 6.31 Contoh Data *Input Corel Data Set*

6.1.2 Informasi *Word* pada data input

Setiap citra mempunyai label 2-3 *word*. Gambar 6.2 menunjukkan frekuensi *word* yang ada dalam keseluruhan citra. Beberapa *word* seperti ‘langit’ mempunyai frekuensi tinggi (sering muncul pada citra), dan *word* ‘matahari’ berfrekuensi rendah (jarang muncul dalam pada citra). Gambar 6.3 menunjukkan frekuensi dari keseluruhan *word* pada data.



Gambar 6.32 Frekuensi *Word* dalam Setiap *Blob*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	0.69759	0.15121	0	0	0	0	0	0	0.15121	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0
4	0	0.25	0	0	0	0	0	0.75	0	0	0	0	0	0
5	0	0	0	0	0	0.25	0	0	0.75	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0
7	0	0	0	0.5	0	0	0	0	0	0.5	0	0	0	0
8	0	1	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0
10	0	0.50042	0	0.4	0	0	0	0	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0.79374	0	0	0	0	0.20626	0	0	0	0	0	0	0
13	0	0.50565	0	0	0.49435	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0.75	0.25	0	0	0
15	0	0	0	0	0	1	0	0	0	0	0	0	0	0
16	0	0	0.66698	0	0	0	0.33302	0	0	0	0	0	0	0
17	0	1	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0
19	1	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	1
21	0	1	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0.33333	0.66667	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0
24	0	1	0	0	0	0	0	0	0	0	0	0	0	0
25	1	0	0	0	0	0	0	0	0	0	0	0	0	0
26	1	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0.16675	0	0.83325	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	1	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	1
30	0	0	0	0	0	0	0	0	0	0	0	0.5	0.5	0
31	0.22222	0.66667	0	0	0	0	0	0	0	0.11111	0	0	0	0
32	0	0.66834	0	0	0	0	0	0.16683	0	0.16683	0	0	0	0
33	0	1	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Gambar 6.33 Frekuensi *Word* dalam Keseluruhan Citra



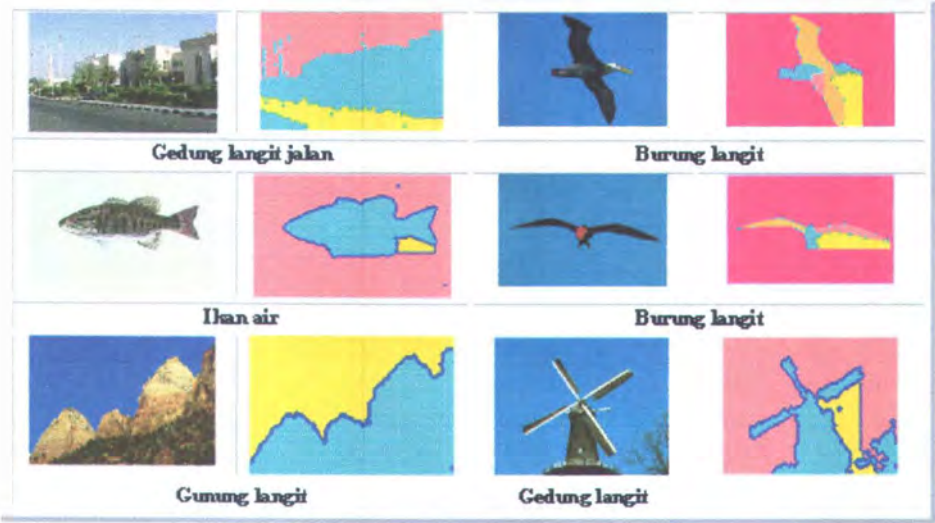
6.2.3 Evaluasi Segmentasi

Penggunaan algoritme *meanshift* dalam proses segmentasi akan mempengaruhi data input citra. Setiap citra disegmentasi dengan algoritme *Mean Shift* pada domain warna  $L*u*v$ .



Gambar 6.34 Perbandingan Hasil Segmentasi *Mean Shift*

Gambar 6.34 menunjukkan hasil segmentasi dari beberapa gambar dengan nilai *spasial* dan nilai *range* yang berbeda untuk satu citra.



Gambar 6.35 Hasil Segmentasi *Mean Shift*

6.2.4 Ekstraksi Fitur Dasar Warna

Serupa dengan algoritme segmentasi yang lain , *Mean Shift* menghasilkan region yang kecil. Dari *n-region* citra berukuran luas > 50 piksel diekstraksi beberapa fitur dasar sebagai berikut:

- 1. Pusat massa  $x, y$
- 2. Area adalah jumlah piksel dalam region :
- 3. Luas *region* dibagi akar dari panjang contour region :
- 4. Mean dan varian dalam domain RGB ,  $L^*u^*v$ ,  $L^*a^*b$ , CrCb

Data citra dikelompokkan dalam 7 kelompok berdasarkan fitur domain warnanya :

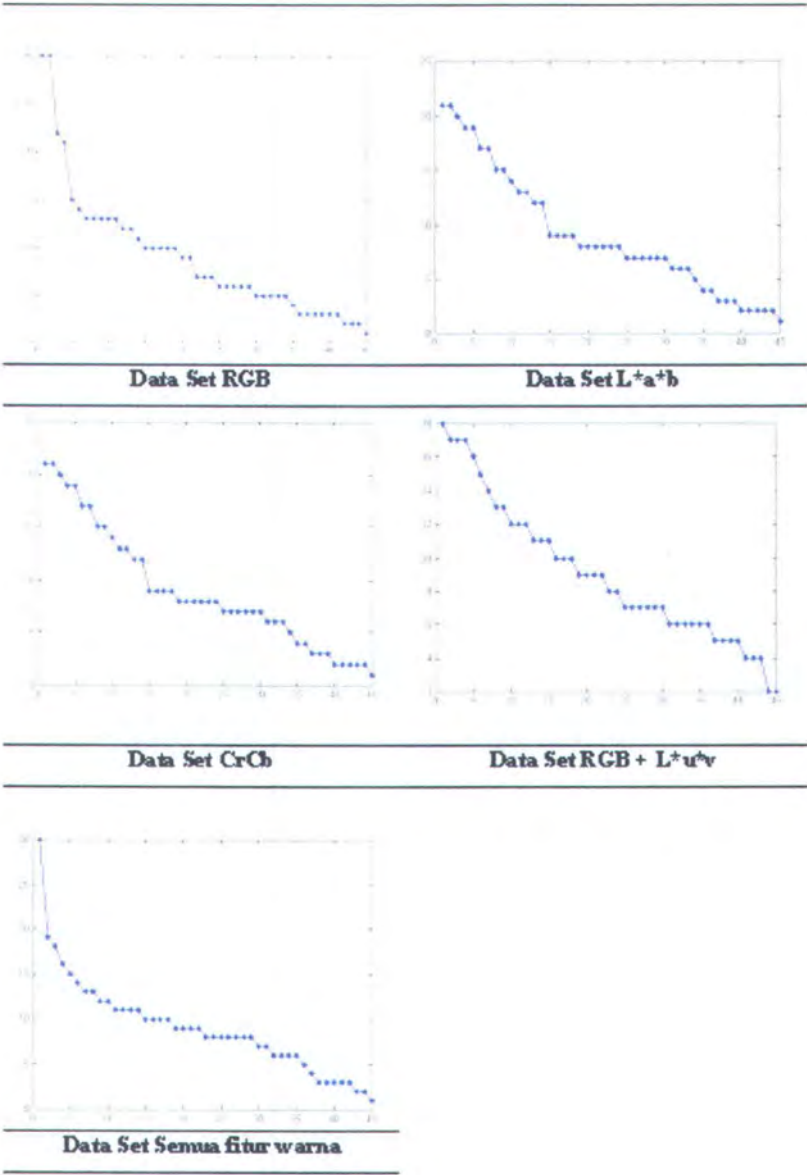
Tabel 6.19 Data set Input

Kelompok Data/dataset	Fitur Warna
1	RGB
2	$L^*u^*v$
3	$L^*a^*b$
4	CrCb
5	RGB dan $L^*u^*v$
6	RGB dan $L^*a^*b$
7	Semua (RGB, $L^*u^*v$ , $L^*a^*b$ , CrCb)

6.2.5 Klasterisasi Region

Nilai  $k$  (jumlah klaster) dipilih berdasarkan keinginan user. Algoritma *K-Means* bukan salah satu algoritme terbaik untuk klasterisasi. Vektor fitur hasil ekstraksi fitur dasar warna dari setiap region citra dinormalisasi dengan distribusi normal dengan *mean* 0 dan varian 1 untuk memetakan vektor fitur dalam *blob*.

Gambar 6.36 menunjukkan distribusi *blob* untuk tahapan pelatihan data dari *dataset* pertama pada 100 jumlah data, dan 55 klaster.



Gambar 6.36 perbandingan klaster antara 5 dataset





Gambar 6.37 *Blob Segmentasi*

Gambar 6.37 menunjukkan *blob* 6 mewakili objek langit pada citra ‘gunung-langit’ , namun juga mewakili objek air pada citra ‘ikan-air’ , pada citra ‘gunung-langit’ objek gunung diwakili pada *blob* 12 dan 18.

Gambar 6.37 menunjukkan *blob* hasil klasterisasi fitur dalam beberapa citra, sebagian besar *region* yang sama terpusat pada kelas yang sama pula. Namun dimungkinkan *region* yang berbeda berada pada kelas yang sama atau fitur yang sama berada pada kelas yang berbeda. Hasil dari pengaruh klasterisasi pada kinerja akan didiskusikan pada sub bab 6.3

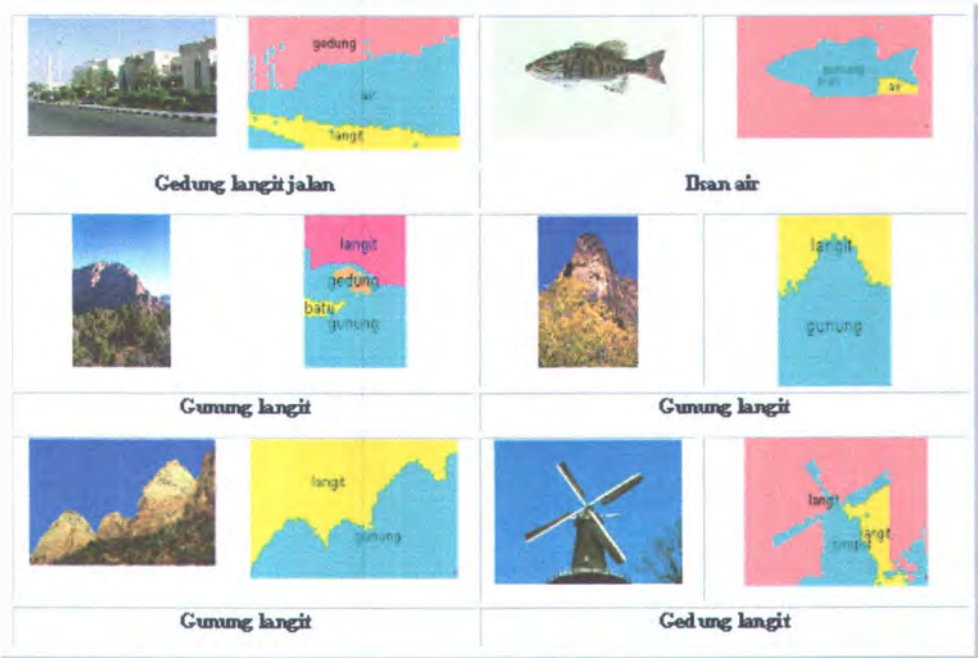


6.2 Metode Evaluasi

Evaluasi strategi *region naming* dijelaskan pada subbab 6.2.1, sedangkan evaluasi strategi *auto-annotation* dijelaskan pada subbab 6.2.2.

6.2.1 Evaluasi Strategi Region Naming

Evaluasi visual adalah hasil dari penggunaan strategi prediksi *region naming*. Setiap *blob* diberi label dengan peluang tertinggi dari prediksi *blob* terhadap *word*. Gambar 6.38 menampilkan hasil dari tahap pelatihan data, beberapa *word* terprediksi dengan benar, dan beberapa *word* yang salah disebabkan karena frekuensi kemunculan *word* tersebut dalam citra.



Gambar 6.38 Hasil strategi *region naming*

### 6.2.2 Evaluasi Strategi *Auto-Annotation*

Strategi *auto-annotation* adalah dengan menjumlahkan semua peluang *blob* terhadap *word* yang terjadi pada citra tersebut sehingga menjadi satu peluang *blob* terhadap *word* yang ternormalisasikan untuk citra tersebut. Label citra tersebut didapatkan dengan mengambil *n*-peluang terbesar dari *blob* terhadap *word* yang ada pada citra, *n* adalah jumlah *keyword* yang ada pada citra.

*Word* prediksi dianggap benar jika *word* tersebut adalah salah satu dari *keyword* yang ada pada citra. Dalam penggunaan strategi ini pengukuran terletak pada kebenaran *word* yang diprediksi bukan pada kebenaran letak *word* prediksi dalam *blob* pada citra. Tiga metode pengukuran *auto-annotation* yaitu Skor *Normalized Clasification*, *Recall* dan *precision*, *True Positive*, *False Positive*, *False Negative* dan memeriksa nilai *maximum likelihood*.

#### 6.2.2.1 Skor *Normalized Clasification*

Terdapat keterbatasan dalam mengevaluasi kinerja prediksi keterhubungan antara *word* dan *blob*. Evaluasi *Auto-Annotation* adalah sebuah proses otomatis dalam pengevaluasian dalam jumlah citra yang besar, diukur dengan membandingkan hasil prediksi *word* dengan *word* yang terlabelkan pada citra.

Skor *Normalized Clasification* yang dikenalkan oleh Barnard [12] :

$$E_{NS} = r/n - w/(N-n) \quad (3.42)$$

*N* : jumlah *word* dalam kamus *word*

*n* : jumlah *word* yang terlabelkan pada citra



$r$  : jumlah *word* yang terprediksi dengan benar

$w$  : jumlah *word* salah

#### 6.2.2.2 *Recall dan precision*

*Recall* dan *precision* didefinisikan pada penilaian *auto- annotation* sebagai berikut:

- ***Recall*** : jumlah *word prediction* yang benar dari seluruh *keyword* yang ada pada citra
- ***Precision*** : jumlah *word* prediksi yang benar dari keseluruhan *word prediction* tersebut muncul.

Kinerja CBIR yang baik ditunjukkan dengan tingginya nilai *recall* dan *precision* dalam kamus *word*. Beberapa *word* mempunyai *recall* dan *precision* yang tinggi. Nilai *recall* tinggi mendekati 1 , berarti *word* diprediksi dengan benar ketika ada pada citra. Jika nilai *precision*-nya kurang dari 0.5 , maka *word* tersebut diprediksi dalam banyak citra, namun kurang dari 0.5 prediksinya benar. Ada beberapa *word* dengan nilai *recall* rendah, namun nilai *precision* tinggi , berarti meskipun *word* tersebut jarang diprediksi, namun seringkali benar.

#### 6.2.2.3 *True Positive, False Positive, False Negative*

Tabel True Positive, False Positive, False Negative menunjukkan kebenaran dari prediksi tiap-tiap *word* dan keseluruhan *word*. *True Positive* adalah jumlah *word* menjadi salah satu *keyword* yang ada pada citra, *true positive* juga dapat digunakan dalam pengukuran kinerja prediksi. *False Positive* adalah

jumlah *word* diprediksi tetapi bukan salah satu dari keyword. *False Negative* adalah jumlah *word* yang menjadi *keyword* tetapi tidak diprediksi.

6.3 Skenario Uji Coba parameter jumlah kelas/*blob*

Skenario uji coba ini dilakukan pada fitur Dataset 4 dengan jumlah klaster yang berbeda. Penentuan nilai *k* dilakukan secara acak. Uji coba ini menganalisis pengaruh parameter jumlah klaster dalam proses perhitungan algoritma *K-Means* pada kinerja CBIR. Tabel 6.20 menunjukkan kinerja dataset pada jumlah klaster yang berbeda. **Jumlah klaster 85 (lebih banyak)** relatif memiliki nilai kinerja yang lebih baik.

Tabel 6.20 Perbandingan iterasi dengan nilai *maximum likelihood*

Jumlah Klaster	N-Skor
45	0.8063
55	0.8316
65	0.8823
75	0.8390
85	0.8895

6.4 Skenario Uji Coba parameter fitur dasar warna

Pada skenario uji coba ini dievaluasi pengaruh parameter pemilihan fitur dasar warna dalam domain warna  $L*u*v$ ,  $L*a*b$ , CrCb, RGB dan  $RGB+L*u*v$ ,  $RGB+L*a*b$  dan gabungan semua domain pada kinerja CBIR.

6.4.1 Uji coba 1

Uji coba pertama dilakukan pada 100 data dengan 7 dataset, 14 *keyword* dan jumlah klaster sebanyak 55. Tabel 6.21 menunjukkan daftar ke-14 *keyword*.

Tabel 6.21 Kamus kata dengan 14 keyword

No Word	Word
1	Gedung
2	Langit
3	Jalan
4	Burung
5	Batu
6	Air
7	Matahari
8	Gunung
9	Ikan
10	Universitas
11	Rumput
12	Pohon
13	alat_tukang
14	Meja

6.4.1.1 Pengukuran Skor *normalized clasification*

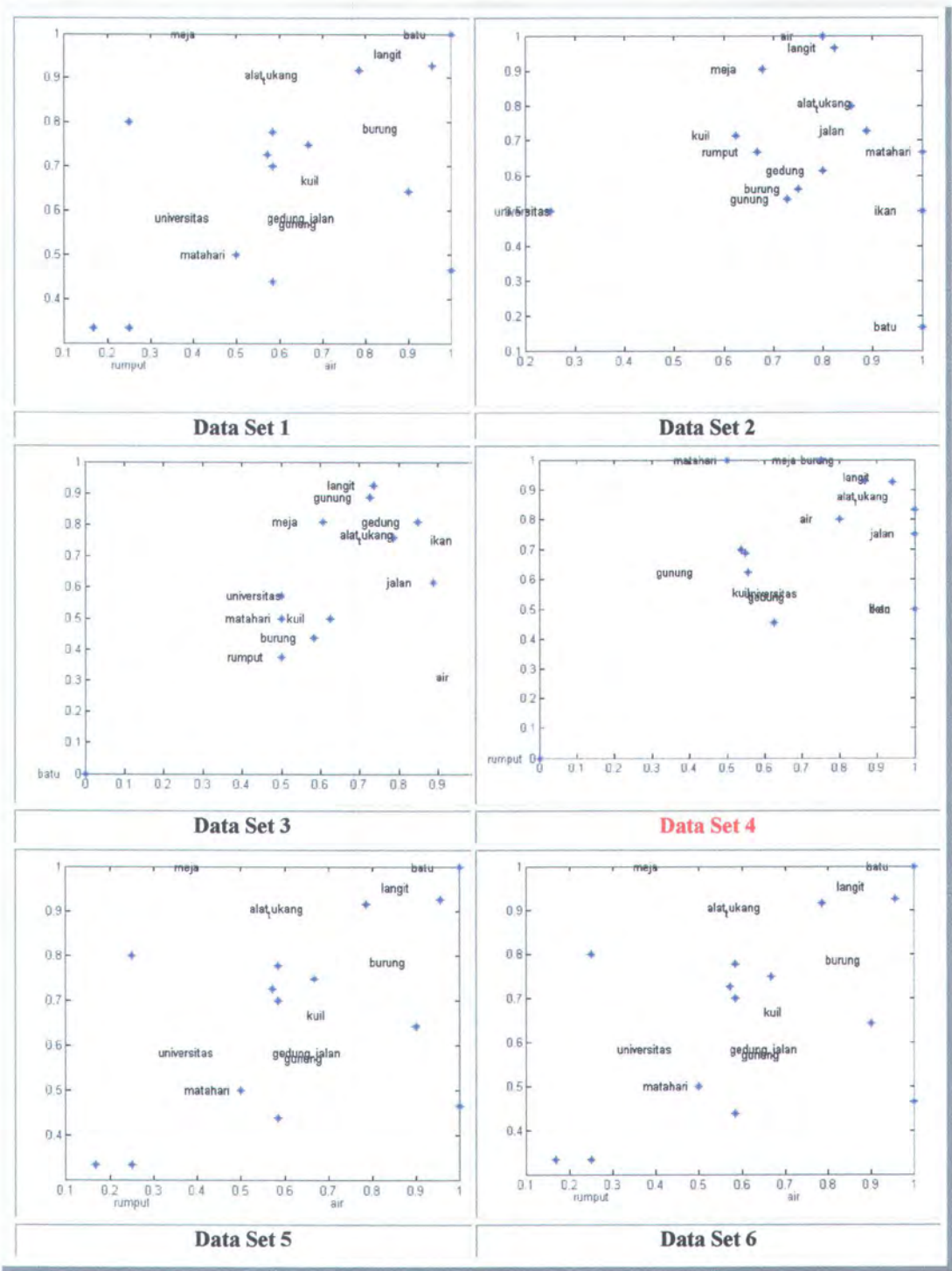
Tabel 6.22 perbandingan *normalized clasification* skor

Data Set	NC Score
1. RGB	0.7571
2. L*a*b	0.7859
3. CrCb	0.7428
4. L*u*v	0.8291
5. RGB + L*u*v	0.7976
6. RGB + L*a*b	0.7518
7. ALL	0.8063

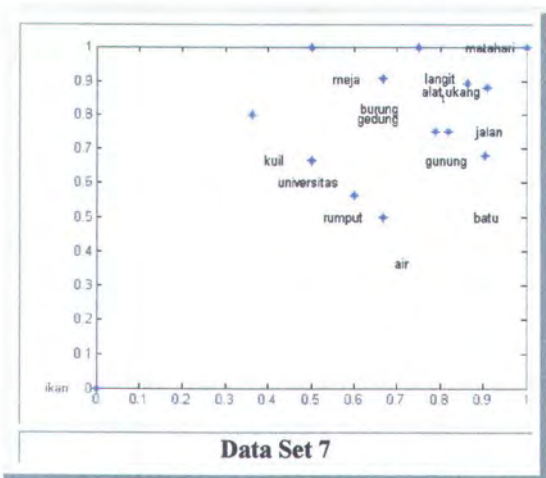
Tabel 6.22 menunjukkan **dataset 4** memiliki nilai *normalized clasification* skor yang tinggi **0.8291** dari dataset yang lain.



6.4.1.2 Pengukuran *Recall/precssion*



Gambar 6.39 *Recall* dan *Precission* dari dataset 1-6



Gambar 6.40 Recall dan Precision dari dataset 7

Pada gambar 6.40 menunjukkan data set 4 memiliki nilai recall /precision > 0.5, yang berarti memiliki kinerja yang lebih baik

6.4.1.3 Pengukuran true postife, word predict , false positif dan false negative

Tabel 6.23 true positive, word predict , false positive dan false negative data set 1-4

No	Data Set 1				Data Set 2				Data Set 3				Data Set 4			
	W_p	Tp	fp	Fn	w_p	tp	fp	fn	w_p	tp	fp	Fn	w_p	tp	fp	fn
1	40	16	24	4	26	16	10	4	21	17	4	3	26	14	12	6
2	71	63	8	5	58	56	2	12	54	50	4	18	67	63	4	5
3	10	5	5	4	11	8	3	1	13	8	5	1	12	9	3	0
4	14	12	2	0	16	9	7	3	16	7	9	5	10	10	0	2
5	2	1	1	0	6	1	5	0	0	0	0	1	2	1	1	0
6	5	3	2	2	4	4	0	1	16	5	11	0	5	4	1	1
7	1	0	1	2	3	2	1	0	2	1	1	1	1	1	0	1
8	16	8	8	3	15	8	7	3	9	8	1	3	8	5	3	6
9	0	0	0	3	6	3	3	0	4	3	1	0	6	3	3	0
10	18	11	7	5	8	4	4	12	14	8	6	8	20	11	9	5
11	2	2	0	4	6	4	2	2	8	3	5	3	0	0	0	6
12	24	22	2	6	30	24	6	4	29	22	7	6	30	26	4	2
13	11	11	0	17	21	19	2	9	21	17	4	11	21	21	0	7
14	3	2	1	6	7	5	2	3	10	5	5	3	9	5	4	3
	217	156			217	163			217	143			217	173		

Tabel 6.24 *true positive, word predict , false positive dan false negative data set 5-7*

No	Data Set 5				Data Set 6				Data Set 7			
	W_p	Tp	fp	fn	w_p	tp	fp	fn	W_p	Tp	fp	fn
1	27	13	14	7	24	14	10	6	19	15	4	5
2	66	59	7	9	66	63	3	5	66	60	6	8
3	9	8	1	1	12	7	5	2	12	9	3	0
4	18	10	8	2	14	11	3	1	11	9	2	3
5	3	1	2	0	1	1	0	0	2	1	1	0
6	6	5	1	0	16	4	12	1	11	4	7	1
7	3	2	1	0	2	1	1	1	2	2	0	0
8	11	8	3	3	14	8	6	3	15	10	5	1
9	1	1	0	2	6	1	5	2	0	0	0	3
10	19	9	10	7	12	7	5	9	15	9	6	7
11	1	1	0	5	8	2	6	4	8	4	4	2
12	26	24	2	4	20	18	2	10	29	25	4	3
13	19	19	0	9	13	13	0	15	21	19	2	9
14	8	5	3	3	9	6	3	2	6	4	2	4
	217	165			217	156			217	171		

Pada tabel 6.23, **dataset 1**, word ke-2 yang mewakili objek **'langit'** memiliki nilai *word predict* 71, *true positive* 63, *false positive* 8, *false negative* 5 (63 *word* yang terprediksi benar, 8 *word* terprediksi salah dari hasil 71 prediksi *word* pada dataset 1 dengan domain warna RGB; 63 *keyword* benar dan 5 *keyword* salah dari 68 *keyword*). Jumlah *keyword* **'langit'** adalah 68 (penjumlahan nilai *true positive* dan *false negative*). Hal ini menunjukkan kinerja sistem dalam memperkirakan kebenaran prediksi word **'langit'** adalah tinggi, karena nilai *true positive* lebih tinggi daripada *false positive* dan *false negative* pada sejumlah *word predict*.

Pada tabel 6.23 dan tabel 6.24 menunjukkan **dataset 4** memiliki nilai kebenaran pada keseluruhan word daripada dataset yang lain. Nilai kebenaran prediksi diperoleh dari jumlah *true positive*.



6.4.2 Uji coba 2

Uji coba pertama dilakukan pada 200 data dengan 7 dataset, 18 *keyword* dan jumlah klaster sebanyak 100. Tabel 6.25 menunjukkan daftar ke-18 *keyword*.

Tabel 6.25 Kamus kata dengan 18 *keyword*

No Word	Word
1	Gedung
2	Langit
3	Jalan
4	Burung
5	Batu
6	Air
7	Matahari
8	Gunung
9	Ikan
10	Universitas
11	Rumput
12	Pohon
13	alat_tukang
14	Meja
15	Kuil
16	Kota
17	Jembatan
18	Tebing

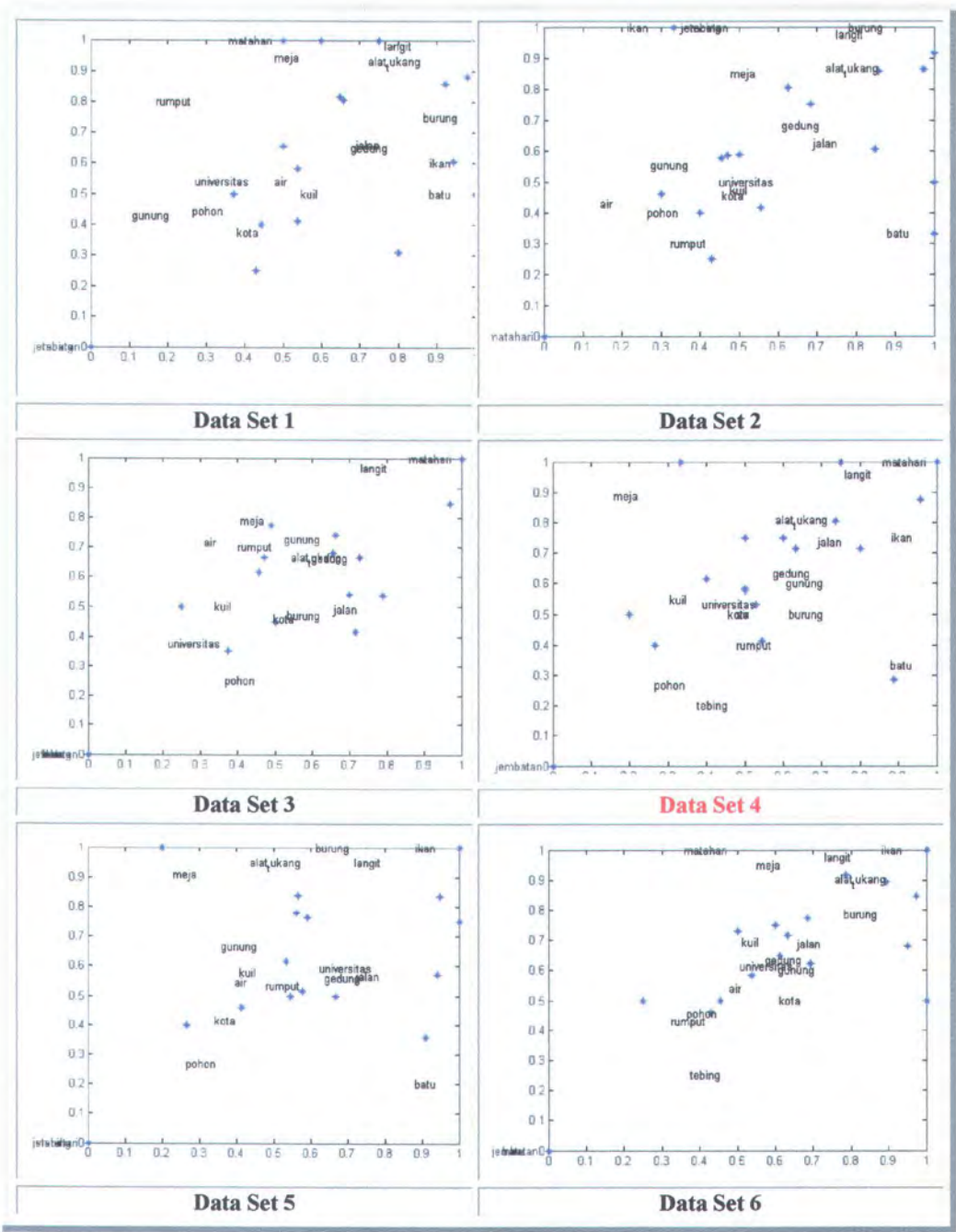
6.4.2.1 Pengukuran Skor *normalized clasification*

Tabel 6.26 perbandingan *normalized clasification* skor

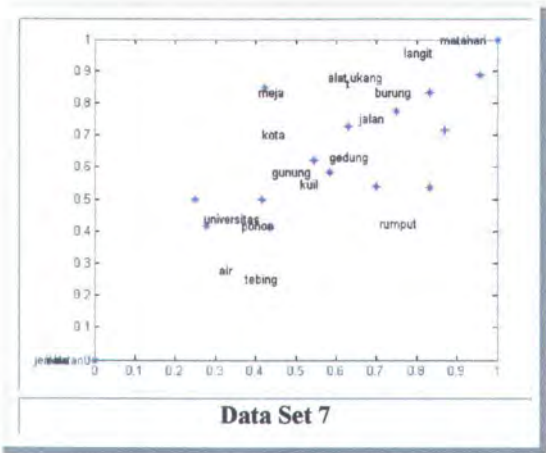
Data Set	Skor <i>normalized clasification</i>
1	0.7704
2	0.7422
3	0.7386
4	0.7794
5	0.7350
6	0.7296
7	0.7526

Tabel 6.26 menunjukkan **dataset 4** memiliki nilai *normalized clasification* skor yang tinggi **0.7794** dari dataset yang lain.

6.4.2.2 Pengukuran *Recall/precision*



Gambar 6.41 *Recall dan Precision* dari dataset 1-6



Gambar 6.42 Recall dan Precision dari dataset 7

Gambar 6.43 menunjukkan **Data set 4** memiliki nilai *recall dan Precision* yang relative lebih dari 0.5

6.4.2.3 Pengukuran true postive, word predict , false positive dan false negative

Pada tabel 6.27, dataset 1, word ke-1 yang mewakili objek '**langit**' memiliki nilai *word predict* 159, *true positive* 156, *false positive* 3, *false negative* 21 (156 *word* yang terprediksi benar, 3 *word* terprediksi salah dari hasil 159 prediksi *word* pada dataset 1 dengan domain warna RGB; 156 *keyword* benar dan 21 *keyword* salah dari 177 *keyword*). Jumlah *keyword* '**langit**' adalah 177 (penjumlahan nilai *true positive* dan *false negative*). Hal ini menunjukkan kinerja sistem dalam memperkirakan kebenaran prediksi word '**langit**' adalah tinggi, karena nilai *true positive* lebih tinggi daripada *false positive* dan *false negative* pada sejumlah *word predict*



Tabel 6.27 true postive, word predict , false positive dan false negative

No Word	Data Set 1				Data Set 2				Data Set 3				Data Set 4			
	wp	tp	fp	fn	wp	tp	fp	fn	wp	tp	fp	fn	wp	tp	fp	Fn
1	97	63	34	14	87	55	32	22	95	62	33	15	85	58	27	19
2	159	156	3	21	162	155	7	22	167	154	13	23	157	153	4	24
3	38	25	13	6	34	25	9	6	43	24	19	7	40	25	15	6
4	16	12	4	0	18	9	9	3	13	8	5	4	11	11	0	1
5	2	1	1	0	3	1	2	0	0	0	0	1	3	1	2	0
6	13	7	6	5	14	7	7	5	11	6	5	6	7	3	4	9
7	1	1	0	1	2	2	0	0	2	2	0	0	0	0	0	2
8	7	3	4	9	15	9	6	3	10	7	3	5	9	5	4	7
9	5	3	2	0	4	3	1	0	0	0	0	3	1	1	0	2
10	13	7	6	10	17	9	8	8	22	9	13	8	20	10	10	7
11	5	4	1	9	20	8	12	5	16	9	7	4	20	6	14	7
12	9	4	5	6	15	4	11	6	18	5	13	5	10	4	6	6
13	26	24	2	4	25	20	5	8	17	13	4	15	28	24	4	4
14	18	17	1	11	9	8	1	20	11	10	1	18	20	17	3	11
15	38	19	19	10	22	12	10	17	28	18	10	11	36	17	19	12
16	35	13	22	13	30	15	15	11	27	14	13	12	33	15	18	11
17	0	0	0	2	0	0	0	2	0	0	0	2	1	1	0	1
18	0	0	0	2	5	1	4	1	2	1	1	1	1	1	0	1
	482	359			482	343			482	342			482	352		

Pada tabel 6.26 dan tabel 6.27 menunjukkan **dataset 5** memiliki nilai kebenaran pada keseluruhan word daripada dataset yang lain. Nilai kebenaran prediksi diperoleh dari jumlah true positive.

Tabel 6.28 true postive, word predict , false positive dan false negative untuk dataset 5-7

No Word	Data Set 5				Data Set 6				Data Set 7			
	Wp	Tp	fp	Fn	wp	tp	fp	fn	Wp	tp	fp	fn
1	87	55	32	22	107	60	47	17	89	56	33	21
2	154	150	4	27	156	148	8	29	164	157	7	20
3	35	24	11	7	46	26	20	5	32	24	8	7
4	14	11	3	1	9	9	0	3	12	10	2	2
5	1	0	1	1	5	1	4	0	0	0	0	1
6	13	7	6	5	11	6	5	6	18	5	13	7
7	1	1	0	1	0	0	0	2	2	2	0	0
8	15	9	6	3	9	6	3	6	12	7	5	5
9	3	3	0	0	3	3	0	0	0	0	0	3
10	18	11	7	6	22	13	9	4	16	7	9	10
11	14	6	8	7	15	8	7	5	26	11	15	2
12	11	5	6	5	15	4	11	6	12	5	7	5
13	28	25	3	3	17	16	1	12	23	20	3	8
14	20	19	1	9	11	10	1	18	18	15	3	13
15	26	18	8	11	26	15	11	14	33	18	15	11
16	38	19	19	7	29	12	17	14	20	14	6	12
17	0	0	0	2	0	0	0	2	1	0	1	2
18	4	1	3	1	1	0	1	2	4	1	3	1
	482	364			482	337			482	352		

Kesimpulan yang diperoleh dari pengukuran parameter fitur domain warna adalah sebagai berikut:

1. Uji Coba 1 : Dengan tingginya nilai kinerja pada **domain L\*u\*v** pada keseluruhan pengukuran *skor normalized clasification*, *tabel true postive - word predict - false positive dan false negative*, maupun *recall/precssison*, maka dapat disimpulkan bahwa pemilihan **domain L\*u\*v** akan memiliki nilai kinerja yang tinggi daripada domain warna yang lain.

2. Uji Coba 2 : domain warna  $L^*u^*v$  memiliki nilai kinerja yang lebih tinggi daripada domain warna yang lain pada pengukuran skor *normalized clasification*, *recall-precission*, namun kombinasi domain warna **RGB dan  $L^*u^*v$**  memperoleh nilai yang lebih tinggi pada pengukuran **tabel *true postive - word predict - false positive dan false negative***.



Yaa ayyatuhan nafsul muthmainnah  
irji'ii ilaa Rabbiki  
raadhiyatan mardhiyyah  
Fadkhulii fii 'ibaadlii  
wadkhulii jannatii

*(Hai jiwa yang tenang  
Kembalilah kamu kepada Tuhanmu  
dengan hati puas lagi diridhai  
Maka masuklah ke dalam golongan hamba-  
hambaKu  
Maka masuklah ke dalam surga-Ku.)*

QS. Al-Fajr: 27-30

## BAB VII KESIMPULAN



## BAB 7

### KESIMPULAN DAN SARAN

Di akhir laporan tugas akhir ini dijelaskan kesimpulan dan kemungkinan pengembangan lebih lanjut dari perangkat lunak yang telah dibuat.

#### 7.1 Kesimpulan

Beberapa kesimpulan yang dapat ditarik dari tugas akhir ini adalah sebagai berikut :

1. **Jumlah penentuan klaster yang lebih banyak** pada klasterisasi K-Means akan **meningkatkan kinerja** sistem temu kembali citra berbasis isi.
  2. Penggunaan fitur dasar warna dalam **domain warna  $L^*u^*v$**  akan **meningkatkan kinerja** sistem temu kembali citra berbasis isi.
- Kesimpulan kedua ini diperoleh dengan menggunakan metode pengukuran skor *normalized clasification* dan *recall-precision*.

#### 7.2 Saran

Agar dapat menghasilkan yang lebih kompleks, pengembangan dapat diarahkan dengan mempertimbangkan hal-hal berikut :

1. Pemilihan nilai parameter pada penggunaan algoritme segmentasi *mean shift* menentukan kinerja sistem. Oleh karena itu diperlukan pengujian dengan parameter yang berbeda pada tahap *pre-processing* sehingga mendapatkan data input citra dengan hasil segmentasi yang baik.

2. Menggunakan metode klasterisasi selain K-Means untuk mendapatkan optimalisasi proses clustering.





# DAFTAR PUSTAKA



## DAFTAR PUSTAKA

- [1] [http://en.wikipedia.org/wiki/Content-based\\_image\\_retrieval](http://en.wikipedia.org/wiki/Content-based_image_retrieval)
- [2] D. Comaniciu, "*Mean Shift*," 2002.
- [3] D. Comaniciu and P. Meer, "*Mean Shift: A Robust Approach toward Feature Space Analysis*," IEEE Trans. Patt. Analy. Mach. Intell., vol. 24, 2002.
- [4] Kobus Barnard, Pinar Duygulu, Raghavendra Guru, Prasad Gabbur, and David Forsyth, " *The Effects of Segmentation and Feature Choice in a Translation Model of Object Recognition* "
- [5] Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik , "*Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying* "IEEE Transactions on pattern analysis and machine intelligence , vol 24 , no.8, august 2002
- [6] Lei Wang, Latifur Khan, "*Automatic Image Annotation and Retrieval Using Weighted Feature Selection* "
- [7] R.O Duda and P.E. hart, pattern Classifiaction and Scene analysis"
- [8] Y.Cheng "Mean Shlft, mode seeking and clustering, IEEE Trans PAMI vol 17 790-799 1995"
- [10] K. Barnard, P. Duygulu, N. de Freitas, D. A. Forsyth, D. Blei, and M. Jordan. "Matching words and pictures". Journal of Machine Learning Research, 3:1107 1135, 2003.